Software Impacts 11 (2022) 100197

Contents lists available at ScienceDirect

Software Impacts

journal homepage: www.journals.elsevier.com/software-impacts

Original software publication

A short tutorial for time series classification and explanation with MrSQM

Thach Le Nguyen *, Georgiana Ifrim

University College Dublin, Ireland

ARTICLE INFO

Keywords:
Time series classification
Feature selection
Python
C++
Linear models
Explanation
Saliency map

ABSTRACT

This paper presents MrSQM, a Python tool for the task of time series classification and explanation. Time series classification is a critical problem not only in scientific research but also in many real-life applications. However, state-of-the-art time series classifiers including deep learning and ensemble architectures are often impractical due to their complexity. MrSQM can provide an alternative lightweight solution, just as accurate but faster, and explainable. The tool is written mainly in C++ but wrapped with Cython to provide a more accessible Python interface.

Code metadata

Current code version	*v0.1*
Permanent link to code/repository used for this code version	https://github.com/SoftwareImpacts/SIMPAC-2021-172
Permanent link to Reproducible Capsule	https://codeocean.com/capsule/3624844/tree/v1
Legal Code License	*GNU General Public License v3.0*
Code versioning system used	*git*
Software code languages, tools, and services used	*C++, Python*
Compilation requirements, operating environments & dependencies	FFTW (http://www.fftw.org/), Cython >= 0.29, numpy >= 1.18, pandas >= 1.0.3, scikit-learn >= 0.22
If available Link to developer documentation/manual	
Support email for questions	thach.lenguyen@ucd.ie

1. Introduction

A time series is a sequence of numerical data values collected over a period of time (e.g., the number of steps a person takes every minute [1]) or based on some other ordering of values such as spatial ordering (e.g., the shape of a coffee leaf or historical artefact [2]). Time series classification is the problem of assigning a class to an unseen time series. Time series data are ubiquitous in almost every aspect of our world and time series databases are some of the fastest growing data systems.[1] Applications of time series classification include human motion classification [3], heart attack detection [4], phoneme recognition [5], earthquake prediction,[2] whale-call detection,[3] and many more. Fig. 1 shows an example of time series data collected by sport scientists at University College Dublin, Ireland. The data was captured using a single accelerometer-based sensor worn on the body by the participants in the study. The participants were asked to perform countermovement jumps (CMJ) while wearing the sensor on their dominant foot. The aim is to assess whether the participants performed CMJ with acceptable technique. More details about this dataset are provided in [3].

A time series classifier is typically deemed useful when it is (1) accurate, (2) efficient, and (3) explainable. Explainability regards the ability

The code (and data) in this article has been certified as Reproducible by Code Ocean: (https://codeocean.com/). More information on the Reproducibility Badge Initiative is available at https://www.elsevier.com/physical-sciences-and-engineering/computer-science/journals.

* Corresponding author.
E-mail address: thach.lenguyen@ucd.ie (T. Le Nguyen).

[1] https://venturebeat.com/2021/01/15/database-trends-the-rise-of-the-time-series-database/.

[2] https://ncedc.org/.

[3] https://www.kaggle.com/c/whale-detection-challenge/data.

https://doi.org/10.1016/j.simpa.2021.100197
Received 29 November 2021; Accepted 29 November 2021

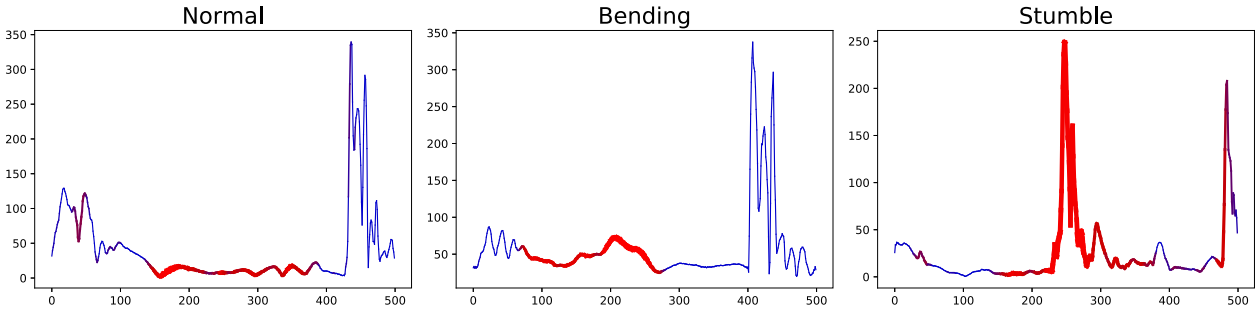

Fig. 1. Accelerometer-based time series data captured from three different executions of the countermovement jump.

to understand the decision-making process of a classifier, e.g., why did the model make this prediction? In the case of time series analysis, a common method of explanation is through a saliency map. A saliency map is basically a vector of weights, one weight for each data point in the time series. The visualization using a saliency map (Fig. 1) highlights the important parts of the time series with regard to the prediction.

It is certainly challenging to achieve all three performance targets. Classic linear models are interpretable and efficient but sometimes not accurate. State-of-the-art deep learning or ensemble architectures are very accurate but often require a tremendous amount of computing resources (e.g., time, memory, space). Moreover, it is not trivial to obtain explanations from such complex models [3,6,7].

MrSQM [8] is a Python tool for time series classification. The core modules of MrSQM are written in C++ and wrapped with Cython for convenience. The tool is fast and accurate. Its performance is comparable to state-of-the-art time series classifiers (e.g., Inception Time [9], TS-CHIEF [10], HIVE-COTE [11], ROCKET [12]). Additionally, it can also provide a saliency map to explain the classification prediction by highlighting the parts of the time series that most influenced the classification decision.

2. Description

2.1. Methodology

Fig. 2 illustrates the architecture of MrSQM. The three main components of MrSQM are: the symbolic transformation module, the feature selection module, and the training module. The symbolic transformation module (SAX [13] or SFA [14]) converts numerical time series to multiple symbolic representations (i.e., sequences of symbols). The feature selection module selects features (in the form of symbolic sequences) from the symbolic representations. The training module (logistic regression) trains a linear classification model for future prediction.

2.2. Implementation

The tool is written in C++ and Python. While we have implemented the algorithm in C++ to maximize the speed, we were aware that Python is more accessible in general. In addition, for the SFA transformation, we reused the C++ code provided by the author of SFA[4] which is significantly faster than its Java and Python alternatives. Therefore we implemented the core modules (the symbolic transformation and feature selection modules) in C++ and wrapped them with Cython. We used *scikit-learn*[5] for model training with logistic regression.

3. Example

In this section, we provide an example of using MrSQM to train and test on a sample dataset.[6] In addition, we show how to obtain the saliency map of a time series for explanation purposes. A more detailed example including the sample dataset can be found in our github repository.[7]

```
1 import mrsqm
2 # Assuming data is loaded in sktime format
3 # https://www.sktime.org/en/stable/examples/
      loading_data.html#sktime_df_format
4 # fit training data
5 clf = mrsqm.MrSQMClassifier().fit(train_X,
      train_y)
6 # make prediction on test data
7 pred_y = clf.predict(test_X)
8 # obtain saliency map for the first time
      series in the test set
9 sm = clf.get_saliency_map(test_X.iloc[0,0])
```

4. Software impact

Research: While machine learning research has had tremendous success and impact through complex deep learning architectures, we still believe that simple methods have their own advantages. MrSQM is part of a group of linear time series classifiers (including WEASEL [15], MrSEQL [3], and ROCKET [12]) that are not only as accurate as their deep learning counterparts, but also usually one or two orders of magnitude faster. By introducing MrSQM, we aim to demonstrate that linear models with their characteristics (fast, interpretable) can also be accurate and thus more suitable for real-life applications.

Furthermore, our method can be easily adapted to time series regression problems. This is showcased in [16], where we successfully applied MrSQM to predict numeric quality traits from milk spectroscopy data in an international data challenge. It is interesting to note that we were the only group in the workshop without a prior background in food spectroscopy and chemometrics. Nonetheless, our experiments with MrSQM produced results that are comparable in accuracy to the traditional approaches in this domain, while also being fast and explainable.

[4] https://www2.informatik.hu-berlin.de/~schaefpa/boss/.
[5] https://scikit-learn.org/stable/.
[6] http://timeseriesclassification.com/description.php?Dataset=Coffee.
[7] https://github.com/mlgig/mrsqm/.

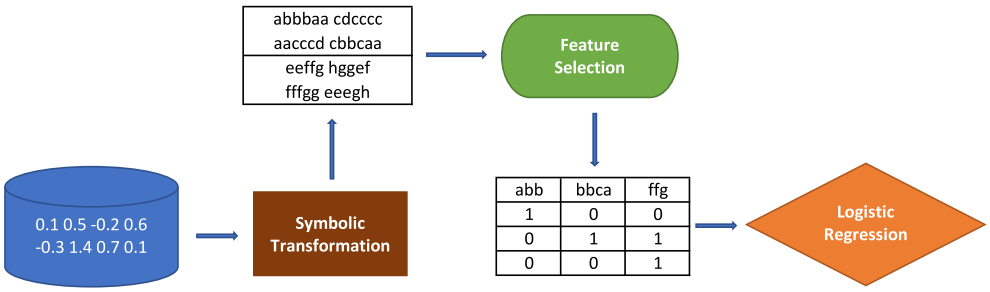

Fig. 2. MrSQM's architecture.

Industry: In our work, we have tested successfully MrSQM with sensor data (Fig. 1). In this type of applications, it is important for the solution to be accurate, fast, lightweight (so it can be deployed on resource-restricted devices like mobile phones), and capable of providing feedback to users (i.e., explanation). Thus we believe MrSQM can be an effective solution to many practical problems with similar requirements.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was funded by Science Foundation Ireland through the VistaMilk SFI Research Centre (SFI/16/RC/3835) and the Insight Centre for Data Analytics, Ireland (12/RC/2289_P2). For the purpose of Open Access, the authors have applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

References

[1] A. Bagnall, H.A. Dau, J. Lines, M. Flynn, J. Large, A. Bostrom, P. Southam, E. Keogh, The UEA multivariate time series classification archive, 2018, 2018, arXiv preprint arXiv:1811.00075.

[2] L. Ye, E. Keogh, Time series shapelets: a novel technique that allows accurate, interpretable and fast classification, Data Min. Knowl. Discov. 22 (1) (2011) 149–182.

[3] 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 Min. Knowl. Discov. 33 (4) (2019) 1183–1222.

[4] R.T. Olszewski, R. Maxion, D. Siewiorek, Generalized feature extraction for structural pattern recognition in time-series data, (Ph.D. thesis), Carnegie Mellon University, USA, 2001, AAI3040489.

[5] H. Hamooni, A. Mueen, Dual-Domain Hierarchical Classification of Phonetic Time Series, in: 2014 IEEE International Conference on Data Mining, 2014, pp. 160–169.

[6] T.T. Nguyen, T. Le Nguyen, G. Ifrim, A model-agnostic approach to quantifying the informativeness of explanation methods for time series classification, in: V. Lemaire, S. Malinowski, A. Bagnall, T. Guyet, R. Tavenard, G. Ifrim (Eds.), Advanced Analytics and Learning on Temporal Data, Springer International Publishing, Cham, 2020, pp. 77–94.

[7] S. Agarwal, T.T. Nguyen, T. Le Nguyen, G. Ifrim, Ranking by aggregating referees: Evaluating the informativeness of explanation methods for time series classification, in: Advanced Analytics and Learning on Temporal Data, 2021.

[8] T.L. Nguyen, G. Ifrim, MrSQM: Fast time series classification with symbolic representations, 2021, https://Arxiv.Org/Abs/2109.01036.

[9] H.I. Fawaz, B. Lucas, G. Forestier, C. Pelletier, D.F. Schmidt, J. Weber, G.I. Webb, L. Idoumghar, P. Muller, F. Petitjean, InceptionTime: Finding AlexNet for time series classification, Data Min. Knowl. Discov. 34 (6) (2020) 1936–1962.

[10] A. Shifaz, C. Pelletier, F. Petitjean, G. Webb, TS-CHIEF: a scalable and accurate forest algorithm for time series classification, Data Min. Knowl. Discov. 34 (2020) 742–775.

[11] J. Lines, S. Taylor, A. Bagnall, HIVE-COTE: The Hierarchical Vote Collective of Transformation-Based Ensembles for Time Series Classification, in: 2016 IEEE 16th International Conference on Data Mining, ICDM, 2016, pp. 1041–1046.

[12] A. Dempster, F. Petitjean, G.I. Webb, ROCKET: exceptionally fast and accurate time series classification using random convolutional kernels, Data Min. Knowl. Discov. 34 (5) (2020) 1454–1495.

[13] J. Lin, E. Keogh, L. Wei, S. Lonardi, Experiencing SAX: a novel symbolic representation of time series, Data Min. Knowl. Discov. 15 (2) (2007) 107–144.

[14] P. Schäfer, M. Högqvist, SFA: A symbolic Fourier approximation and index for similarity search in high dimensional datasets, in: Proceedings of the 15th International Conference on Extending Database Technology, in: EDBT, vol. 12, ACM, New York, NY, USA, 2012, pp. 516–527, URL http://doi.acm.org/10.1145/2247596.2247656.

[15] P. Schäfer, U. Leser, Fast and accurate time series classification with WEASEL, in: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, in: CIKM, vol. 17, ACM, New York, NY, USA, 2017, pp. 637–646, URL http://doi.acm.org/10.1145/3132847.3132980.

[16] M. Frizzarin, A. Bevilacqua, B. Dhariyal, K. Domijan, F. Ferraccioli, E. Hayes, G. Ifrim, A. Konkolewska, T. Le Nguyen, U. Mbaka, G. Ranzato, A. Singh, M. Stefanucci, A. Casa, Mid infrared spectroscopy and milk quality traits: A data analysis competition at the "international workshop on spectroscopy and chemometrics 2021", Chemometr. Intell. Lab. Syst. 219 (2021) 104442.