The Journal of Open Source Software

CategoricalTimeSeries.jl: A toolbox for categorical time-series analysis

Corentin Nelias[1, 2]

1 Max Planck Institute for Dynamics and Self-Organization **2** Department of Physics, Georg-August-Universität Göttingen

DOI: 10.21105/joss.03733

Software

- Review
- Repository
- Archive

Editor: Mehmet Hakan Satman

Reviewers:

- @bkamins
- @felixcremer

Submitted: 11 September 2021
Published: 02 November 2021

Introduction

CategoricalTimeSeries.jl is a Julia toolbox made for analysing categorical time-series.

Categorical time-series are time-sequenced data in which the values at each time point are categories rather than measurements. The common approach to deal with categorical time-series consists in transforming the data via a mapping to obtain a real-valued sequence. This enables the use of traditional time-series analysis methods. However, most of these methods (power-spectral density estimation, correlation coefficients, dimensionality reduction, etc.) are not invariant under general transformations and will produce different results based on the choice of mapping. Therefore, depending on the type of categorical data and the problem at hand, it is desirable to have methods that work with the direct categorical values themselves.

The purpose of **CategoricalTimeSeries.jl** is to provide such tools. The package comes with extensive documentation available online: https://categoricaltimeseriesjl.readthedocs.io/en/latest/

Statement of need

While several implementations of categorical time-series analysis methods are already available, they are written in different languages, some of which are not free (e.g., Matlab). Additionally, no implementations for methods such as the *spectral envelope* or the *random projection* (see *Overview of functionality* below) are available online. This package centralizes and implements most of the standard methods of categorical time-series analysis in a single toolbox fully written in the Julia language.

Overview of functionality

This toolbox was designed to be easy to use and to produce results that are simple to plot. Consequently, the methods implemented in the package take the inputs as 1-D arrays of any type. Type conversion and pre-processing (when needed) are done automatically within the methods without the need for additional coding by the user. The results are either formatted in a way that can be plotted directly with the `Plots.jl` library, or a helper function is provided for visualization and interpretation.

The main areas of functionality are:

Spectral analysis: The spectral envelope method (Stoffer et al., 1993) is used to study the power-spectrum of categorical time-series. As stated in the Introduction section, the

power-spectrum of a time-series is not invariant under a generic transformation. A wrong choice of mapping can potentially flatten certain peaks and render them unnoticeable. For each frequency, the spectral envelope seeks the mapping that maximizes the value of the power-spectrum normalized by the total variance. The `spectral_envelope` function takes a time-series (1-D array) as the input and returns all the frequencies of the spectrum and the values of the intensity associated with the optimal mappings. It also returns the mappings. For a finer study of the mappings themselves, the `get_mappings` function can be used, instead.

Association analysis: The notion of auto-correlation function is not formally defined for a categorical time-series (Weiß, 2018). Yet it might be of interest to know how interdependent the values of the time-series are. We implemented several coefficients generalizing the concept of linear correlations to categorical time-series. Cramer's coefficient, Cohen's coefficient, and Theil's U can be computed via the `cramer_coefficient`, `cohen_coefficient`, and `theils_u` functions, respectively. They take a 1-D array representing the time-series to study and an array of lags storing the lag values at which the coefficients are evaluated as the inputs.

Motif recognition: Time-series can present repeating motifs that are worthwhile identifying. However, simple line-search algorithms are not adapted for all motifs (Pevzner et al., 2000). Moreover, the lack of proper distance measurement complicates the search in the context of categorical time-series.
An implementation using the *random projection* method (Buhler & Tompa, 2002) is used here. The identification of potential motifs is performed by the `detect_motifs` function. It takes a time-series (1-D array), the length of the motifs to look for, and the number of allowed errors as input arguments. It returns an instance of the `pattern` structure which stores properties of the identified motif such as shapes, repetition number, and positions.

Data clustering: If certain categories in a time-series present functional similarities, one might wish to cluster them together into a single equivalent representation. This reduces the total number of categories and can simplify the analysis of the time-series. For this purpose, we use an implementation based on the *Information bottleneck* concept (Strouse & Schwab, 2017; Tishby et al., 2000). After an initial bottleneck model of the structure `IB` is instantiated, it can be optimized with the `IB_optimize!` function to reveal potential clusters of categories. An overview of the results can be obtained with the `print_results` function.

Acknowledgements

The author thanks Nori Jacoby for discussing and providing insight on the *Information bottleneck* concept.

References

Buhler, J., & Tompa, M. (2002). Finding motifs using random projections. *Journal of Computational Biology*, *9*(2), 225–242. https://doi.org/10.1145/369133.369172

Pevzner, P. A., Sze, S.-H., & others. (2000). Combinatorial approaches to finding subtle signals in DNA sequences. *ISMB*, *8*, 269–278. ISBN: 978-0-493-55010-7

Stoffer, D. S., Tyler, D. E., & McDougall, A. J. (1993). Spectral analysis for categorical time series: Scaling and the spectral envelope. *Biometrika*, *80*(3), 611–622. https://doi.org/10.1093/biomet/80.3.611

Strouse, D., & Schwab, D. J. (2017). The deterministic information bottleneck. *Neural Computation*, *29*(6), 1611–1630. https://doi.org/10.1162/neco_a_00961

Tishby, N., Pereira, F. C., & Bialek, W. (2000). The information bottleneck method. *arXiv Preprint Physics/0004057*.

Weiß, C. H. (2018). *An introduction to discrete-valued time series*. John Wiley & Sons. https://doi.org/10.1002/9781119097013