The Journal of Open Source Software

ewstools: A Python package for early warning signals of bifurcations in time series data

Thomas M. Bury [1,2]

1 Department of Physiology, McGill University, Montréal, Canada **2** Department of Applied Mathematics, University of Waterloo, Waterloo, Canada

DOI: 10.21105/joss.05038

Software

- Review
- Repository
- Archive

Editor: Øystein Sørensen

Reviewers:

- @mhu48
- @mikesha2
- @ranzhengcode

Submitted: 13 December 2022
Published: 11 February 2023

License
Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

Summary

Many systems in nature and society have the capacity to undergo critical transitions: sudden and profound changes in dynamics that are hard to reverse. Examples include the outbreak of disease, the collapse of an ecosystem, and the onset of a cardiac arrhythmia. From a mathematical perspective, these transitions may be understood as the crossing of a bifurcation (tipping point) in an appropriate dynamical system model. In 2009, Scheffer and colleagues proposed early warning signals (EWS) for bifurcations based on statistics of noisy fluctuations in time series data (Scheffer et al., 2009). This spurred massive interest in the subject, resulting in a multitude of different EWS for anticipating bifurcations (Clements & Ozgul, 2018). More recently, EWS from deep learning classifiers have outperformed conventional EWS on several model and empirical datasets, whilst also providing information on the type of bifurcation (Bury et al., 2021). Software packages for EWS can facilitate the development and testing of EWS, whilst also providing the scientific community with tools to rapidly apply EWS to their own data.

`ewstools` is an accessible Python package for computing, analysing, and visualising EWS in time series data. The package provides:

- An intuitive, object-oriented framework for working with EWS in a given time series
- A suite of temporal EWS and associated methods (Dakos et al., 2012)
- A suite of spectral EWS (Bury et al., 2020)
- Methods to use deep learning classifiers for EWS (Bury et al., 2021)
- Integrated plotting and evaluation functions to quickly check performance of EWS
- Built-in theoretical models to test EWS
- Interactive tutorials in the form of Jupyter notebooks

`ewstools` makes use of several open-source Python packages, including pandas (McKinney, 2010; The pandas development team, 2020) for dataframe handling, NumPy (Harris et al., 2020) for fast numerical computing, Plotly (Plotly Technologies Inc., 2015) for visualisation, LMFIT (Newville et al., 2016) for nonlinear least-squares minimisation, ARCH (Sheppard, 2015) for bootstrapping methods, statsmodels (Seabold & Perktold, 2010) and SciPy (Virtanen et al., 2020) for detrending methods, and Keras (Chollet & others, 2015) and TensorFlow (Abadi et al., 2016) for deep learning.

Statement of need

Critical transitions are relevant to many disciplines, including ecology, medicine, finance, and epidemiology, to name a few. As such, it is important that EWS are made widely accessible. To my knowledge, there are two other software packages developed for computing EWS, namely earlywarnings by Dakos et al. (2012) and spatialwarnings by Génin et al. (2018),

The Journal of Open Source Software

which both use the R programming language. Given the recent surge in popularity of the Python programming language (PYPL, 2022), a Python-based implementation of EWS should be useful. `ewstools` also implements novel deep learning methods for EWS, which have outperformed conventional EWS in several model and empirical systems (Bury et al., 2021). These new methods should be tried, tested, and developed for a variety of systems and I hope that this package facilitates this endeavour.

Usage Example

```python
import ewstools

# Load data and get time series as a pandas Series object
df = pd.read_csv('data.csv')
series = df['x']

# Initialise ewstools TimeSeries object and define transition time
ts = ewstools.TimeSeries(data=series, transition=440)

# Detrend time series
ts.detrend(method='Lowess', span=0.2)

# Compute desired EWS
ts.compute_var(rolling_window=0.5)
ts.compute_auto(lag=1, rolling_window=0.5)
ts.compute_auto(lag=2, rolling_window=0.5)

# Compute performance metrics
ts.compute_ktau()

# Plot results - can be saved as an interactive html file or as a static image
fig = ts.make_plotly()
```

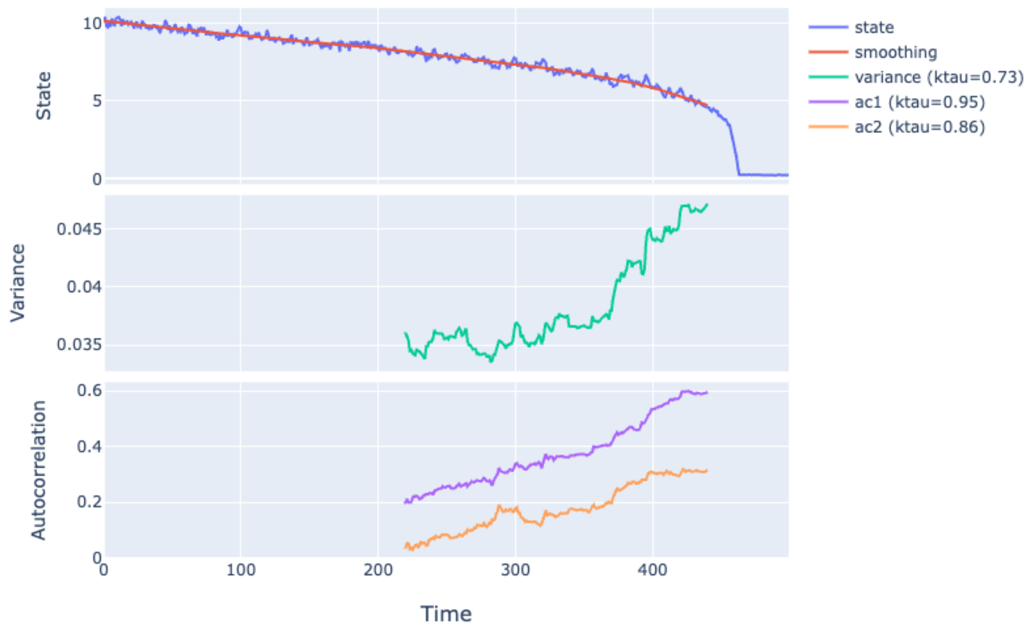

Figure 1: Output of plotting function in usage example.

Documentation

Documentation for `ewstools` is available at https://ewstools.readthedocs.io/en/latest/. Tutorials in the form of Jupyter notebooks are available at https://github.com/ThomasMBury/ewstools/tree/main/tutorials.

Acknowledgements

This work is supported by the Fonds de Recherche du Québec – Nature et technologies (FRQNT) and Compute Canada. Earlier versions were supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., & others. (2016). TensorFlow: A system for large-scale machine learning. *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 265–283. https://doi.org/10.48550/arXiv.1605.08695

Bury, T. M., Bauch, C. T., & Anand, M. (2020). Detecting and distinguishing tipping points using spectral early warning signals. *Journal of the Royal Society Interface*, *17*(170), 20200482. https://doi.org/10.1098/rsif.2020.0482

Bury, T. M., Sujith, R., Pavithran, I., Scheffer, M., Lenton, T. M., Anand, M., & Bauch, C. T. (2021). Deep learning for early warning signals of tipping points. *Proceedings of the National Academy of Sciences*, *118*(39), e2106140118. https://doi.org/10.1073/pnas.2106140118

Chollet, F., & others. (2015). *Keras*. https://github.com/fchollet/keras; GitHub.

Clements, C. F., & Ozgul, A. (2018). Indicators of transitions in biological systems. *Ecology Letters*, *21*(6), 905–919. https://doi.org/10.1111/ele.12948

Dakos, V., Carpenter, S. R., Brock, W. A., Ellison, A. M., Guttal, V., Ives, A. R., Kéfi, S., Livina, V., Seekell, D. A., van Nes, E. H., & others. (2012). Methods for detecting early warnings of critical transitions in time series illustrated using simulated ecological data. *PloS One*, *7*(7), e41010. https://doi.org/10.1371/journal.pone.0041010

Génin, A., Majumder, S., Sankaran, S., Danet, A., Guttal, V., Schneider, F. D., & Kéfi, S. (2018). Monitoring ecosystem degradation using spatial data and the R package spatialwarnings. *Methods in Ecology and Evolution*, *9*(10), 2067–2075. https://doi.org/10.1111/2041-210X.13058

Harris, C. R., Millman, K. J., Van Der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., & others. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

McKinney, Wes. (2010). Data Structures for Statistical Computing in Python. In Stéfan van der Walt & Jarrod Millman (Eds.), *Proceedings of the 9th Python in Science Conference* (pp. 56–61). https://doi.org/10.25080/Majora-92bf1922-00a

Newville, M., Stensitzki, T., Allen, D. B., Rawlik, M., Ingargiola, A., & Nelson, A. (2016). *Lmfit: Non-linear least-square minimization and curve-fitting for Python* (p. ascl:1606.014). Astrophysics Source Code Library, record ascl:1606.014.

Plotly Technologies Inc. (2015). *Collaborative data science*. Plotly Technologies Inc. https://plotly.com

PYPL. (2022). *Popularity of programming language index*. https://pypl.github.io/PYPL.html

Scheffer, M., Bascompte, J., Brock, W. A., Brovkin, V., Carpenter, S. R., Dakos, V., Held, H., Van Nes, E. H., Rietkerk, M., & Sugihara, G. (2009). Early-warning signals for critical transitions. *Nature*, *461*(7260), 53–59. https://doi.org/10.1038/nature08227

Seabold, S., & Perktold, J. (2010). Statsmodels: Econometric and statistical modeling with Python. *9th Python in Science Conference*. https://doi.org/10.25080/Majora-92bf1922-011

Sheppard, K. (2015). *ARCH toolbox for Python*. Zenodo. https://doi.org/10.5281/zenodo.15681

The pandas development team. (2020). *Pandas-dev/pandas: pandas* (latest) [Computer software]. Zenodo. https://doi.org/10.5281/zenodo.3509134

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., & others. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, *17*(3), 261–272. https://doi.org/10.1038/s41592-020-0772-5