The Journal of Open Source Software

pactus: A Python framework for trajectory classification

G. Viera-López [1*¶], J. J. Morgado-Vega [2*], A. Reyes [3*], E. Altshuler [3], Yudivián Almeida-Cruz [2], and Giorgio Manganini [1]

1 Department of Computer Science, Gran Sasso Science Institute, L'Aquila, Italy **2** Department of Artificial Intelligence, University of Havana, Havana, Cuba **3** Group of Complex Systems and Statistical Physics, University of Havana, Havana, Cuba ¶ Corresponding author * These authors contributed equally.

DOI: 10.21105/joss.05738

Software

- Review
- Repository
- Archive

Editor: Arfon Smith

Reviewers:

- @JustinShenk
- @miladmzdh

Submitted: 08 June 2023
Published: 22 September 2023

Summary

Trajectory classification concerns the forecasting of the class or category of a mobile entity based on its observed motion over time. It is a problem that has been studied in many different fields, including robotics, behavior analysis, mobility pattern mining, and user activity recognition (Silva et al., 2019). This task presents multiple challenges for conventional classification models, such as the indeterminate length of trajectories (Li et al., 2022), the range of entities that generate trajectories (Bae et al., 2022; Janczura et al., 2020; Xiao et al., 2017), and the absence of established standards in trajectory datasets (Bae et al., 2022; Xiao et al., 2017).

Our study endeavors to lay the foundation for the assessment of innovative methods and extend their outcomes to a broader range of datasets. We introduce a new framework, referred to as `pactus`, which addresses the challenges of trajectory classification by providing direct access to a carefully chosen collection of datasets and several trajectory classifiers. `pactus` facilitates researchers' ability to experiment with various approaches and assess their performance on different types of data. A comprehensive software documentation is provided on (https://pactus.readthedocs.io/en/latest/).

Statement of need

In recent years, several software libraries have emerged, aiming to automate trajectory data analysis. Within the R community, there are various available tools (Joo et al., 2020). Recognizing the popularity and extensive usage of Python, the `traja` software (Shenk et al., 2021) was developed to integrate different analysis techniques for two-dimensional trajectories, primarily focusing on animal behavioral analysis. Additionally, the `yupi` library (Reyes et al., 2023) was created to handle trajectory analysis for applications involving an arbitrary number of dimensions.

Although these libraries offer valuable tools for trajectory classification, such as classification models and feature extraction from trajectories, they were not specifically designed for this task. Consequently, contemporary research on trajectory classification faces limitations in terms of evaluation, often considering only a limited number of datasets or reporting only a reduced set of metrics (Bae et al., 2022).

The lack of standardization in trajectory datasets, coupled with the difficulty of obtaining these datasets for evaluation, poses a significant challenge to researchers working in fields related to trajectory classification. Moreover, the absence of a reliable and reproducible evaluation methodology makes it difficult to compare different methods and assess their performance accurately. There is a pressing need for a standardized framework to facilitate the evaluation of trajectory classification models in an open and transparent fashion.

In response to these challenges, we have developed `pactus`, a novel framework that provides researchers with direct access to a curated list of datasets and relevant trajectory classifiers. `pactus` offers a user-friendly interface that encourages researchers to include their own datasets or methods on the platform, thereby expanding the range of datasets available for evaluation. The library's evaluation methodology ensures that results are reproducible and comparable, making it easier to identify the most effective trajectory classification methods for specific scenarios. Finally, it encourages the production of reproducible research by enabling researchers to distribute their findings as simple Python scripts, relying on `pactus` for all tasks related to data acquisition, processing, and model evaluation.

Pactus Software Library

The functionalities of `pactus` can be divided into modules: Data handling, Feature extraction, Classification models and Evaluation.

Data handling

The library provides direct access to some of the most commonly used datasets for trajectory classification. The selection of datasets was conducted with meticulous care to encompass a broad range of trajectories and classification objectives. Our initial selection includes GeoLife (Zheng et al., 2009, 2008, 2010), The Starkey Project dataset, also known as `Animals` in the trajectory classification community (Rapp, 2009), four different datasets from the the UCI repository (Dua & Graff, 2017) and two different hurricane datasets, provided by National Hurricane Center (Landsea & Franklin, 2013) and the China Meteorological Administration (Lu et al., 2021; Ying et al., 2014) respectively. To ensure consistency, all datasets were transformed into a standardized format utilizing the trajectory data structures proposed in (Reyes et al., 2023). Datasets are not bundled with the software package, but rather will be downloaded and cached automatically upon each individual access through the library. A complete guide on how to use custom datasets or requesting the inclusion of new datasets into `pactus` can be found in the documentation.

Feature extraction

In order to mitigate the different-length trajectories on some datasets, `pactus` is able to extract statistical features from any trajectory and convert an arbitrary length trajectory into a fixed size vector whose components are engineered features typically used in the literature (Xiao et al., 2017; Zheng et al., 2008).

Users can implement their own method to perform this conversion, and an example on how to do it can be found in the documentation. However, there is a default method that uses all the features computed by the `yupi` library.

Classification models and Evaluation

Several classification algorithms are included in `pactus`. Some of them can be evaluated on the vectorized versions of the trajectories (e.g., Random Forest, SVM, KNN). In other cases the classifiers are able to handle variable-size inputs (e.g., LSTM or Transformers (Bae et al., 2022)) and can be evaluated directly on the trajectory data. In both cases, typical evaluation metrics for classification are computed automatically for the model being evaluated.

Overview

All the functionalities of the library can be integrated in a single script. Figure 1 shows an example on how to use `pactus` for training and evaluating a Random Forest model using the Starkey Project dataset, also known as `Animals`.

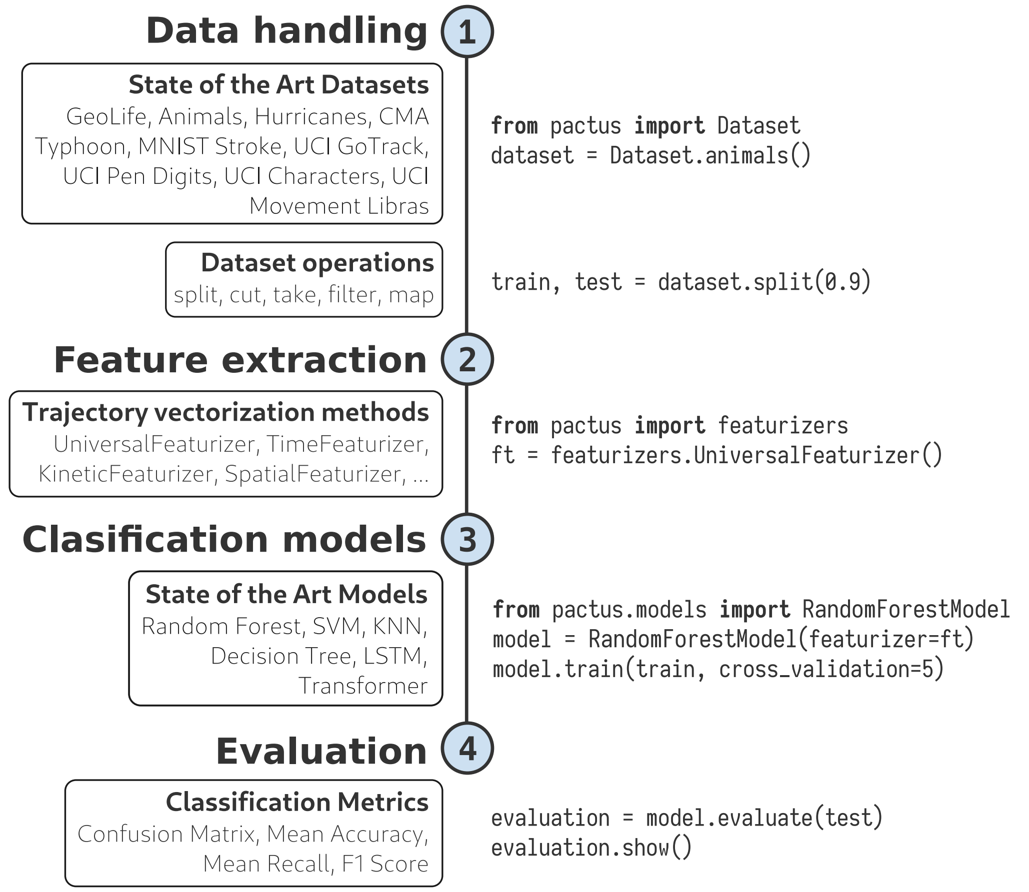

Figure 1: Overview of the resources available in `pactus` coupled with an usage example.

Conclusions

The software presented with this work, `pactus`, addresses typical challenges faced in trajectory classification research. By providing researchers with direct access to curated datasets and trajectory classifiers, `pactus` enhances the availability of resources for evaluation. It is concieved with extensibility in mind, encouraging researchers to contribute their own datasets and methods. The evaluation methodology ensures reproducibility and comparability of results, facilitating the identification of effective trajectory classification methods for specific scenarios. Additionally, pactus promotes reproducible research by enabling researchers to distribute their findings as Python scripts, relying on pactus for data acquisition, processing, and model evaluation. Overall, pactus offers a valuable tool for researchers in the field of trajectory classification, addressing key challenges and facilitating future advancements in the field.

References

Bae, K., Lee, S., & Lee, W. (2022). Transformer networks for trajectory classification. *2022 IEEE International Conference on Big Data and Smart Computing (BigComp)*, 331–333.

https://doi.org/10.1109/bigcomp54360.2022.00074

Dua, D., & Graff, C. (2017). *UCI machine learning repository*. University of California, Irvine, School of Information; Computer Sciences. http://archive.ics.uci.edu/ml

Janczura, J., Kowalek, P., Loch-Olszewska, H., Szwabiński, J., & Weron, A. (2020). Classification of particle trajectories in living cells: Machine learning versus statistical testing hypothesis for fractional anomalous diffusion. *Physical Review E*, *102*(3), 032402. https://doi.org/10.1103/physreve.102.032402

Joo, R., Boone, M. E., Clay, T. A., Patrick, S. C., Clusella-Trullas, S., & Basille, M. (2020). Navigating through the r packages for movement. *Journal of Animal Ecology*, *89*(1), 248–267. https://doi.org/10.1111/1365-2656.13116

Landsea, C. W., & Franklin, J. L. (2013). Atlantic hurricane database uncertainty and presentation of a new database format. *Monthly Weather Review*, *141*(10), 3576–3592. https://doi.org/10.1175/mwr-d-12-00254.1

Li, Q., He, X., Chen, K., & Ouyang, Q. (2022). A two-stage semi-supervised high maneuvering target trajectory data classification algorithm. *Applied Sciences*, *12*(21), 10979. https://doi.org/10.3390/app122110979

Lu, X., Yu, H., Ying, M., Zhao, B., Zhang, S., Lin, L., Bai, L., & Wan, R. (2021). Western north pacific tropical cyclone database created by the china meteorological administration. *Advances in Atmospheric Sciences*, *38*(4), 690–699. https://doi.org/10.1007/s00376-020-0211-7

Rapp, V. (2009). *Elk, deer, and cattle: The starkey project*. DIANE Publishing.

Reyes, A., Viera-López, G., Morgado-Vega, J. J., & Altshuler, E. (2023). Yupi: Generation, tracking and analysis of trajectory data in python. *Environmental Modelling & Software*, *163*, 105679. https://doi.org/10.1016/j.envsoft.2023.105679

Shenk, J., Byttner, W., Nambusubramaniyan, S., & Zoeller, A. (2021). Traja: A python toolbox for animal trajectory analysis. *Journal of Open Source Software*, *6*(63), 3202. https://doi.org/10.21105/joss.03202

Silva, C. L. da, Petry, L. M., & Bogorny, V. (2019). A survey and comparison of trajectory classification methods. *2019 8th Brazilian Conference on Intelligent Systems (BRACIS)*, 788–793. https://doi.org/10.1109/bracis.2019.00141

Xiao, Z., Wang, Y., Fu, K., & Wu, F. (2017). Identifying different transportation modes from trajectory data using tree-based ensemble classifiers. *ISPRS International Journal of Geo-Information*, *6*(2), 57. https://doi.org/10.3390/ijgi6020057

Ying, M., Zhang, W., Yu, H., Lu, X., Feng, J., Fan, Y., Zhu, Y., & Chen, D. (2014). An overview of the china meteorological administration tropical cyclone database. *Journal of Atmospheric and Oceanic Technology*, *31*(2), 287–301. https://doi.org/10.1175/JTECH-D-12-00119.1

Zheng, Y., Li, Q., Chen, Y., Xie, X., & Ma, W.-Y. (2008). Understanding mobility based on GPS data. *Proceedings of the 10th International Conference on Ubiquitous Computing*, 312–321. https://doi.org/10.1145/1409635.1409677

Zheng, Y., Xie, X., Ma, W.-Y., & others. (2010). GeoLife: A collaborative social networking service among user, location and trajectory. *IEEE Data Eng. Bull.*, *33*(2), 32–39.

Zheng, Y., Zhang, L., Xie, X., & Ma, W.-Y. (2009). Mining interesting locations and travel sequences from GPS trajectories. *Proceedings of the 18th International Conference on World Wide Web*, 791–800. https://doi.org/10.1145/1526709.1526816