The Journal of Open Source Software

traffic, a toolbox for processing and analysing air traffic data

Xavier Olive[1]

1 ONERA, Université de Toulouse, France

DOI: 10.21105/joss.01518

Software

- Review
- Repository
- Archive

Submitted: 20 June 2019
Published: 05 July 2019

Summary

Problems tackled by researchers and data scientists in aviation and air traffic management (ATM) require manipulating large amounts of data representing trajectories, flight parameters and geographical descriptions of the airspace they fly through.

Trajectories are mathematical objects used to describe the evolution of a moving object with a finite list of parameters. The most common features in ATM include latitude, longitude, altitude, all indexed by time, with derived ones such as ground speed, track angle and vertical rate. Aircraft dynamic models may expect pitch, roll and yaw angles, together with indicated, computed or true airspeed, Mach number, and more.

Further, airspaces are a key element of aviation: they are regulated by specific rules, whereby navigation is allowed to determined types of aircraft meeting strict requirements. Such volumes, assigned to air traffic controllers to ensure the safety of flights and proper separation between aircraft are most commonly described as a combination of extruded polygons.

Common operations relevant to trajectories evolving in controlled airspaces range from basic attributes: time of entry, time of exit, duration, maximum or minimum altitudes or speed; to more complex operations like intersections of trajectories with airspaces, distances between pairs of trajectories and more. Top performance and expressivity are key expectations for common tasks like preprocessing and filtering of data, preparation of trajectory datasets, or computation of key performance indicators.

The `traffic` library uses the Python language to reach a large base of academics and data scientists users, to serve and benefit its active community and to build on top of a large catalogue of libraries. Trajectories are modelled on top of Pandas dataframe, a natural solution to represent time series, while airspaces leverage Shapely (Gillies & others, 2007) geometries and operations (intersections, inclusion, cascaded joins among others).

`traffic` provides key operations for analysing trajectories evolving in airspaces. It is particularly useful to programmers and researchers needing to compute statistics, performance indicators and building datasets for common machine learning tasks. Meaningful methods are efficiently written using Pandas and Shapely optimised methods, more obvious operations are directly passed to the underlying dataframes.

Core structure of the library

`traffic` acts as declarative grammar designed to preprocess collections of trajectories represented as `core.Traffic` classes holding a Pandas DataFrame as a single attribute. `core.Traffic` provides indexing on `core.Flight` structures, unfolds map operations—transforming

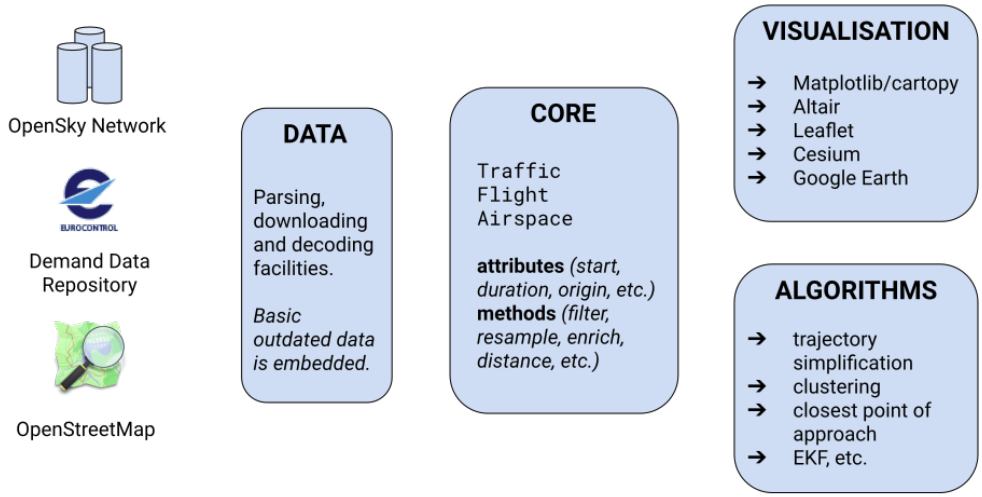

Figure 1: Architecture of the library

a `core.Flight` into another `core.Flight`—, and filtering operations—associating a boolean to a `core.Flight`, set to False if the trajectory should be discarded.

Methods on `core.Flight` are designed to be chained in a similar way Pandas functions. For the sake of expressivity, any method returning a `core.Flight` or a boolean can be stacked on `core.Traffic` structures, in order to be later lazily executed into one single multiprocessed iteration.

Basic navigational data are embedded in the library, together with parsing facilities for most common sources of information, with a main focus on Europe at the time being. A particular attention has been put to a proper binding to ADS-B and Mode S data accessible through the OpenSky (Schäfer, Strohmeier, Lenders, Martinovic, & Wilhelm, 2014) infrastructure, which is openly accessible to academics; decoding capability for stored raw messages is also provided via pyModeS library (Sun, Vû, Ellerbroek, & Hoekstra, 2019)

Finally, `traffic` library provides facilities to export data in common popular visualisation tools including Matplotlib/Cartopy (Elson et al., 2018), Google Earth, Leaflet, Cesium JS for geographical coordinates and altair (VanderPlas et al., 2018) for other features. For the time being, any other kind of visualisation can be implemented as a plugin.

Related works

`traffic` has already been used as a preprocessing tool to prepare data for data analytics (Schäfer et al., 2019) and machine learning methods (Olive & Bieber, 2018, Olive, Grignard, Dubot, & Saint-Lot (2018), Olive & Basora (2019), Olive & Morio (2019)).

Several third party users have already been using and contributing feedback to the library, mostly as bug reports and minor bug fixes.

We expect to improve the library with optimisations in processing time, enriched algorithms and more powerful visualisation capabilities.

References

Elson, P., Andrade, E. S. de, Hattersley, R., Campbell, E., Dawson, A., May, R., scmc72, et al. (2018, November). SciTools/cartopy: v0.17.0. doi:10.5281/zenodo.1490296

Gillies, S., & others. (2007). Shapely: Manipulation and analysis of geometric objects. toblerity.org. Retrieved from https://github.com/Toblerity/Shapely

Olive, X., & Basora, L. (2019). Identifying Anomalies in past en-route Trajectories with Clustering and Anomaly Detection Methods. In *Proceedings of the Air Traffic Management Seminar*.

Olive, X., & Bieber, P. (2018). Quantitative Assessments of Runway Excursion Precursors using Mode S data. In *Proceedings of the International Conference for Research in Air Transportation*.

Olive, X., & Morio, J. (2019). Trajectory clustering of air traffic flows around airports. *Aerospace Science and Technology*, *84*, 776–781. doi:10.1016/j.ast.2018.11.031

Olive, X., Grignard, J., Dubot, T., & Saint-Lot, J. (2018). Detecting Controllers' Actions in Past Mode S Data by Autoencoder-Based Anomaly Detection. In *Proceedings of the SESAR Innovation Days* (p. 8).

Schäfer, M., Olive, X., Strohmeier, M., Smith, M., Martinovic, I., & Lenders, V. (2019). OpenSky Report 2019: Analysing TCAS in the Real World using Big Data. In *Proceedings of the 38th Digital Avionics Systems Conference (DASC)*.

Schäfer, M., Strohmeier, M., Lenders, V., Martinovic, I., & Wilhelm, M. (2014). Bringing up OpenSky: A large-scale ADS-B sensor network for research. In *Proceedings of the 13th International Symposium on Information Processing in Sensor Networks*. doi:10.1109/IPSN.2014.6846743

Sun, J., Vû, H., Ellerbroek, J., & Hoekstra, J. M. (2019). PyModeS: Decoding Mode S Surveillance Data for Open Air Transportation Research. *IEEE Transactions on Intelligent Transportation Systems*. doi:10.1109/TITS.2019.2914770

VanderPlas, J., Granger, B., Heer, J., Moritz, D., Wongsuphasawat, K., Satyanarayan, A., Lees, E., et al. (2018). Altair: Interactive Statistical Visualizations for Python. *Journal of Open Source Software*. doi:10.21105/joss.01057