SeqTools: a python package for easy transformation, combination and evaluation of large datasets

Nicolas Granger, Mounîm a El Yacoubi

The Journal of Open Source Software

SeqTools: A python package for easy transformation, combination and evaluation of large datasets.

Nicolas Granger[1] **and Mounîm A. El Yacoubi**[1]

1 Télécom SudParis

DOI: 10.21105/joss.01006

Software

- Review
- Repository
- Archive

Submitted: 05 October 2018
Published: 26 October 2018

Summary

`SeqTools` facilitates the manipulation of large datasets and the evaluation of a transformation pipeline. Some of the provided functionnalities include: mapping element-wise operations, reordering, reindexing, concatenation, joining, slicing, minibatching, etc…

To improve ease of use, `SeqTools` assumes that dataset are objects that implement a list-like sequence interface: a container object with a length and its *elements accessible via indexing or slicing*. All `SeqTools` functions take and return objects compatible with this simple and convenient interface.

Sometimes manipulating a whole dataset with transformations or combinations can be slow and resource intensive; a transformed dataset might not even fit into memory! To circumvent this issue, `SeqTools` implements *on-demand* execution under the hood, so that computations are only run when needed, and only for actually required elements while ignoring the rest of the dataset. This helps to keep memory resources down to a bare minimum and accelerate the time it take to access any arbitrary result. This on-demand strategy helps to quickly define dataset-wide transformations and probe a few results for debugging or prototyping purposes, yet it is transparent for the users who still benefit from a simple and convenient list-like interface.

When comes the transition from prototyping to execution, the list-like container interface facilitates serial evaluation. Besides, `SeqTools` also provides simple helpers to dispatch work between multiple background workers (threads or processes), and therefore to maximize execution speed and resource usage.

`SeqTools` originally targets data science, more precisely the preprocessing stages of a dataset. Being aware of the experimental nature of this usage, on-demand execution is made as transparent as possible to users by providing fault-tolerant functions and insightful error reporting. Moreover, internal code is kept concise and clear with comments to facilitate error tracing through a failing transformation pipeline.

Nevertheless, this project purposedly keeps a generic interface and only requires minimal dependencies in order to facilitate reusability beyond this scope of application.

Related Work

Joblib, proposes low-level functions with many optimization settings to optimize pipelined transformations. This library notably provides advanced caching mechanisms which are not the primary concern of `SeqTool`. `SeqTool` uses a simpler container-oriented interface with multiple utility functions in order to assist fast prototyping. On-demand evaluation

is its default behaviour and applies at all layers of a transformation pipeline. In particular, parallel evaluation can be inserted in the middle of the transformation pipeline and won't block the execution to wait for the computation of all elements from the dataset.

`SeqTools` is conceived to connect nicely to the data loading pipeline of Machine Learning libraries such as (Paszke et al., 2017) or (Abadi et al., 2015). The interface of these libraries focuses on iterators to access transformed elements, contary to `SeqTools` which also provides arbitrary reads via indexing.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., et al. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Retrieved from https://www.tensorflow.org/

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., et al. (2017). Automatic differentiation in pytorch. In *NIPS-w*.