The Journal of Open Source Software

Museo ToolBox: A Python library for remote sensing including a new way to handle rasters.

Nicolas Karasiak[1]

1 Université de Toulouse, INRAE, UMR DYNAFOR, Castanet-Tolosan, France

DOI: 10.21105/joss.01978

Software

- Review
- Repository
- Archive

Editor: Katy Barnhart

Reviewers:

- @cmillion
- @mollenburger

Submitted: 12 December 2019
Published: 21 April 2020

Summary

`Museo ToolBox` is a Python library dedicated to the processing of georeferenced arrays, also known as rasters or images in remote sensing.

In this domain, classifying land cover type is a common and sometimes complex task, regardless of your level of expertise. Recurring procedures such as extracting Regions Of Interest (ROIs, or raster values from a polygon), computing spectral indices or validating a model with a cross-validation can be difficult to implement.

`Museo ToolBox` aims at simplifying the whole process by making the main treatments more accessible (extracting of ROIs, fitting a model with cross-validation, computing Normalized Difference Vegetation Index (NDVI) or various spectral indices, performing any kind of array function to the raster, etc).

The main objective of this library is to facilitate the transposition of array-like functions into an image and to promote good practices in machine learning.

To make `Museo ToolBox` easier to get started with, a full documentation with lot of examples is available online on read the docs.

Museo ToolBox in details

`Museo ToolBox` is organized into several modules (Figure 1):

- processing: raster and vector processing.
- cross-validation: stratified cross-validation compatible with scikit-learn.
- ai: artificial intelligence module built upon scikit-learn (Pedregosa et al., 2011).
- charts: plot confusion matrix with F1 score or producer/user's accuracy.
- stats: compute statistics (such as Moran's Index (Moran, 1950), confusion matrix, commision/omission) or extracting truth and predicted label from a confusion matrix.

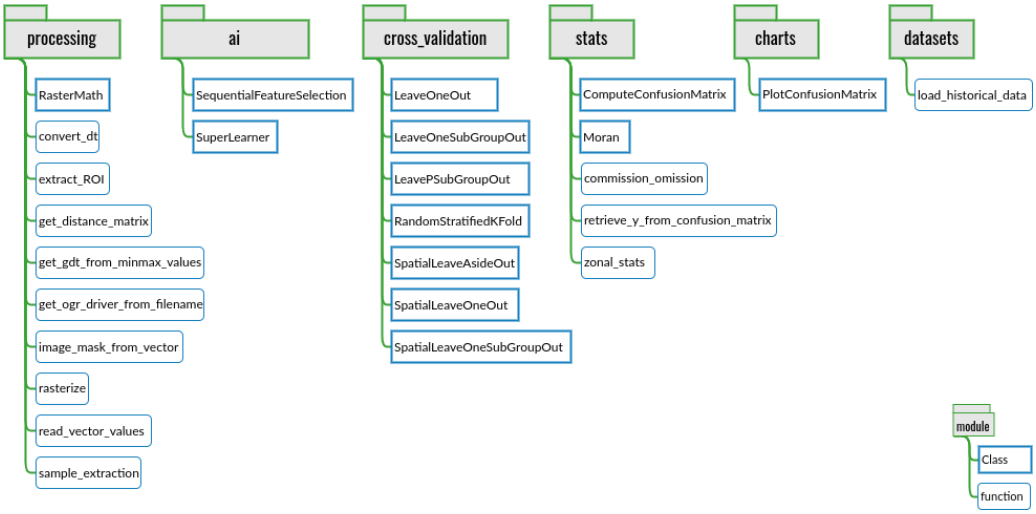

Figure 1: Museo ToolBox schema.

The main usages of `Museo ToolBox` are:

1. Reading and writing a raster block per block using your own function.
2. Generating cross-validation, including spatial cross-validation.
3. Fitting models with scikit-learn, extracting accuracy from each cross-validation fold, and predicting raster.
4. Plotting confusion matrix and adding f1 score or producer/user accuracy.
5. Getting the y_true and and y_predicted labels from a confusion matrix.

RasterMath

Available in `museotoolbox.processing`, the `RasterMath` class is the keystone of `Museo ToolBox`.

The question I asked myself is: How can we make it as easy as possible to implement array-like functions on images? The idea behind `RasterMath` is that if the function is intended to operate with an array, it should be easy to use it with your raster using as few lines as possible.

So, what does `RasterMath` really do? The user only works with an array and confirms with a sample that the process is doing well, and lets `RasterMath` generalize it to the whole image. The user doesn't need to manage the raster reading and writing process, the no-data management, the compression, the number of bands, or the projection. Figure 2 describes how `RasterMath` reads a raster, performs the function, and writes it to a new raster.

The objective is to **allow the user to focus solely on the array-compatible function** while `RasterMath` manages the raster part.

See `RasterMath` documentation and examples.

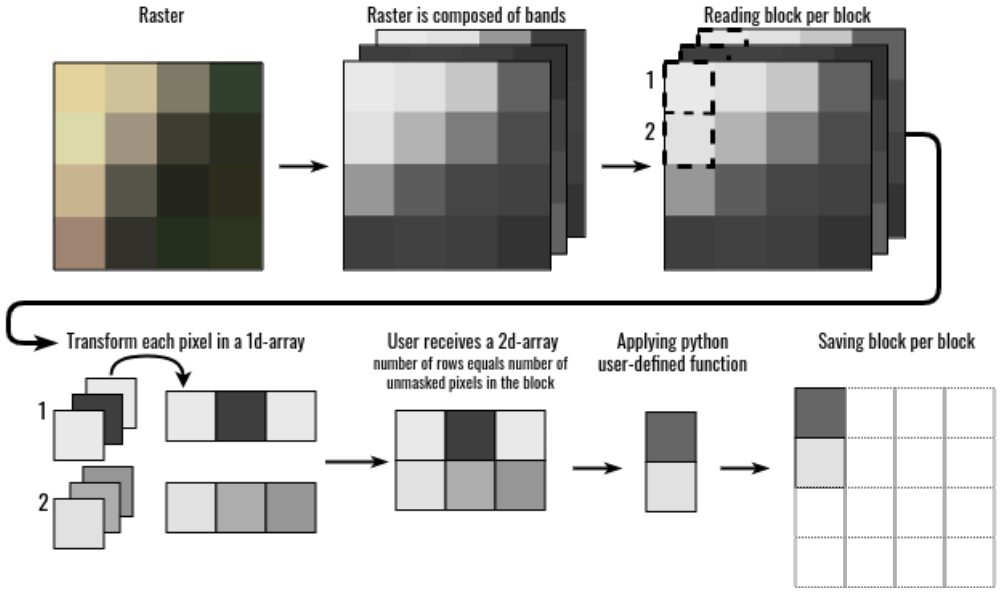

Figure 2: `RasterMath` under the hood

Artificial Intelligence

The artificial intelligence (`ai`) module is natively built to implement `scikit-learn` algorithms and uses state of the art methods (such as standardizing the input data). `SuperLearner` class optimizes the fit process using a grid search to fix the parameters of the classifier. There is also a Sequential Feature Selection protocol which supports a number of components (e.g. a single-date image is composed of four bands, i.e. four features, so a user may select four features at once).

See the `SuperLearner` documentation and examples.

Cross-validation

`Museo ToolBox` implements stratified cross-validation, which means the separation between the training and the validation samples is made by respecting the size per class. For example the Leave-One-Out method will keep one sample of validation per class. As stated by Olofsson et al. (2014) *"stratified random sampling is a practical design that satisfies the basic accuracy assessment objectives and most of the desirable design criteria"*. For spatial cross-validation, see Karasiak et al. (2019) inspired by Roberts et al. (2017).

`Museo ToolBox` offers two different kinds of cross-validation:

Non-spatial cross-validation

- Leave-One-Out.
- Leave-One-SubGroup-Out.
- Leave-P-SubGroup-Out (Percentage of subgroup per class).
- Random Stratified K-Fold.

Spatial cross-validation

- Spatial Leave-One-Out (Karasiak et al., 2019).

- Spatial Leave-Aside-Out.
- Spatial Leave-One-SubGroup-Out (using centroids to select one subgroup and remove other subgroups for the same class inside a specified distance buffer).

See the cross-validation documentation and examples.

Acknowledgements

I acknowledge contributions from Mathieu Fauvel, beta-testers (hey Yousra Hamrouni), and my thesis advisors: Jean-François Dejoux, Claude Monteil and David Sheeren. Many thanks to Marie for proofreading. Many thanks to Sigma students: Hélène Ternisien de Boiville, Arthur Duflos, Sam Antonetti and Anne-Sophie Tronc for their involvement in `RasterMath` improvements in early 2020.

References

Karasiak, N., Dejoux, J.-F., Fauvel, M., Willm, J., Monteil, C., & Sheeren, D. (2019). Statistical stability and spatial unstability in prediction of forest tree species using satellite image time series. *Remote Sensing*. doi:10.3390/rs11212512

Moran, P. A. P. (1950). Notes on Continuous Stochastic Phenomena. *Biometrika*, *37*(1/2), 17–23. doi:10.2307/2332142

Olofsson, P., Foody, G. M., Herold, M., Stehman, S. V., Woodcock, C. E., & Wulder, M. A. (2014). Good practices for estimating area and assessing accuracy of land change. *Remote Sensing of Environment*, *148*, 42–57. doi:10.1016/j.rse.2014.02.015

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., et al. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Roberts, D. R., Bahn, V., Ciuti, S., Boyce, M. S., Elith, J., Guillera-Arroita, G., Hauenstein, S., et al. (2017). Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure. *Ecography*, *40*(8), 913–929. doi:10.1111/ecog.02881