GeoPy: A Python Package for Geospatial and Statistical Analysis

Andre R. Erler

November 4, 2015

The GeoPy Analysis Package

The GeoPy Python package was developed to facilitate data analysis in a workstation environment. It is intended for two major use cases: scripted post-processing of datasets and interactive data analysis. The package is comprised of several sub-packages that provide functionality at different stages of the analysis, as outlined below.

The scripted post-processing capability is primarily intended for automatic post-processing of monthly WRF data (output produced by the WRFavg package), but also to pre-process observational and other datasets. It is fully parallelized and provide comprehensive logging and debugging options.

The interactive data analysis functionality is primarily designed to provide a unified abstraction layer for datasets and take advantage of available meta data to facilitate slicing and dicing operations, reprojection/regridding, and statistical and time series operations. The plotting package is also fully aware of these meta data and automatically handles much

of the plot annotation and layout. It also provides methods for many common types of plots, but collection is by no means complete. The data analysis functionality provided in GeoPy is meant to be relatively generic and extensible. For specific projects it is recommended to build a specialized package on top of GeoPy, as has been done for this work with the EVA and Hydro packages. The preferred front-end to work with GeoPy in an exploratory analysis context is the IPython notebook. A set of commented IPython notebooks containing most of the analysis work flow for this dissertation can be found in the code repositories associated with the EVA and Hydro packages.

A short description of the major sub-packages and their primary classes and functions follows; documentation for all modules, classes, methods, and functions is available in the sources code (in doc-strings and comments):

`geodata` This sub-package contains the core data structure, as well as statistical functions for time series analysis (based on SciPy) and functionality for geo-spatial analysis, such as regridding (based on GDAL. It implements Variable and Dataset classes for data abstraction and a container class to facilitate batch operations (`base.py`), as well as child classes that provide access to NetCDF datasets and variables (`netcdf.py`). The `gdal.py` module implements a projection class, as well as functions to add geo-spatial functionality to Variables and Datasets. A variety of statistical tests and functions are implemented in the `stats.py` module, along with a new distribution variable class (VarRV) with integrated support for bootstrapping and cross-validation. The latter is a container class for SciPy's `rv_continuous` class, in the same way as the other Variable classes are containers for NumPy arrays and netCDF-4 Variables. The statistical operations provided in this package are capable of operating on multi-dimensional arrays and are fully parallelized;[1] other

[1]Unfortunately the vast majority of statistical functions in the SciPy `stats.py` module are not vectorized, so that a universal wrapper had to be developed, to apply statistical functions to a multi-dimensional

operations, such as regridding or basic arithmetics, are fully vectorized but not parallelized.

`datasets` A collection of wrapper modules for specific datasets. In order to access a dataset through the unified abstraction layer provided by GeoPy, a wrapper module has to be added here and imported to access the data. The following datasets are currently supported: CESM (including CVDP data; see *Erler*, 2015, §2.1.5), WRF, CFSR, NARR, CRU, GPCC, PRISM, as well as river gauge station (`WSC`; see *Erler*, 2015, §2.1.3) and meteorological station observations (`EC`; see *Erler*, 2015, §2.1.4), and the unified/merged observations (`Unity`). The `common.py` module also contains helper routines to facilitate loading of multiple datasets and automatic expansion of argument lists.

`processing` This sub-package contains a module implementing basic batch processing, including comprehensive logging and parallelization, based on the multiprocessing package. Built on top of these modules are batch processing scripts for the following applications: extraction of station locations from fields (`exstns.py`), regridding (`regrid.py`), area averages based on shape files (`shpavg.py`), and climatological means from monthly WRF output (`wrfavg.py`). All operations in this module are fully parallelized on a dataset and file type basis.

`plotting` The primary functionality of this sub-package is implemented by subclassing the Figure and Axes classes of the 2D plotting library matplotlib; the subclasses MyFigure and MyAxes are drop-in replacements for their parent classes and implement line plot and histogram methods that are fully aware of GeoPy container classes and their meta data and extended functionality. Various helper methods and functions, e.g. to construct confidence intervals from bootstrapping and add annotation, are also provided. The Axes class currently does not support GeoPy-aware surface or vector plots, but an extension to

array; parallelization of statistical functions is achieved by parallelization within the wrapper.

that effect, based on the cartopy-package is planned. Currently, a capable but convoluted legacy script is used for this purpose (`multimap.py`), which relies on the older basemap-package. Variable properties and default settings are defined in separate modules. The package also contains a collection of colormaps and style sheets that were mostly adapted from other packages.

`projects` This is a container for project specific modules that should probably be moved into a separate repository. Among other scripts, it also contains a registry of all WRF simulations.

`utils` These modules extend existing NumPy and SciPy functions or provide generic utility functions like EOF analysis. They do not reference GeoPy data structures and can be used independently.

`atmdyn` This module was developed for the PyGeode packages.[2] Adaptation for GeoPy is planned, but still outstanding. The core of this package is a library containing functions to compute common meteorological quantities, such as Potential Vorticity (PV) written in Python or Fortran (with OpenMP parallelization). Python wrappers for the Fortran library are available for PyGeode, but have not been adapted for use with GeoPy.

In addition, unittest modules with over 100 tests are provided that test most of the key functionality.

The package is continuously being developed and improved; a recent version is available on GitHub: `https://github.com/aerler/GeoPy`

[2] PyGeode is a Python package originally co-developed by Dr. P. Hitchcock, M. Neish with contributions from the author; it is available on GitHub: `https://github.com/pygeode/pygeode`

References

Erler, A. R. (2015), High Resolution Hydro-climatological Projections for Western Canada, Ph.D. thesis, University of Toronto.