Scikit-Learn: Machine Learning in the Python ecosystem

Gilles Louppe University of Liège, Belgium
Gaël Varoquaux Parietal, INRIA Saclay, France

Scikit-Learn

The *scikit-learn*[1][2] project [4] is an increasingly popular machine learning library written in Python. It is designed to be simple and efficient, useful to both experts and non-experts, and reusable in a variety of contexts. The primary aim of the project is to provide a compendium of efficient implementations of classic, well-established machine learning algorithms. Among other things, it includes classical supervised and unsupervised learning algorithms, tools for model evaluation and selection, as well as tools for data preprocessing and feature engineering. *scikit-learn* is distributed under the 3-clause BSD license, encouraging its free use in both commercial and academic settings.

Started in 2007, *scikit-learn* is developed by an international team of over a dozen core developers, mostly researchers from various fields of science. The project also benefits from many occasional contributors proposing small bugfixes or improvements. Development proceeds on GitHub, which greatly facilitates this kind of collaboration. Because of the large number of developers, emphasis is put on keeping the project maintainable. Code must follow quality guidelines, such as style consistency and unit-test coverage. Documentation and examples are required for all features, and major changes must pass code review by developers not involved in the proposed change.

All algorithms within *scikit-learn* are offered through a simple and elegant API [1] consisting of a well-defined set of methods. This API consistency across the package makes it very usable in practice: experimenting with different learning algorithm is as simple as substituting a class definition. Through composition interfaces, the library also offers powerful mechanisms to express a wide variety of learning tasks within a small amount of easy-to-read code. Finally, through duck-typing, the consistent API leads to a library that is easily extensible, and allows user-defined estimators to be incorporated into the *scikit-learn* workflow without any explicit object inheritance.

[1] `http://scikit-learn.org`

[2] `http://mloss.org/software/view/240`

Integration in the Python ecosystem

The library has been designed to tie in with standard open source tools of the scientific Python ecosystem. In particular, *scikit-learn* leverages NumPy [6] for efficient storage and manipulation of multi-dimensional arrays, and SciPy [3] for more specialized data structures (e.g. sparse matrices) and implementations of lower-level scientific algorithms. The scikit-learn API is designed to avoid the proliferation of framework code: it is limited and non-intrusive. As such it makes *scikit-learn* easy to use and easy to combine with other libraries. Together with IPython [5] for interactive exploration and Matplotlib [2] for dynamic data visualization, NumPy and SciPy constitute a comprehensive scientific working environment that *scikit-learn* smoothly complements with a host of machine learning algorithms and data analysis routines.

Demonstrations

This presentation will illustrate the use of *scikit-learn* as a component of the larger scientific Python environment to solve complex data analysis tasks. Examples will include end-to-end workflows based on powerful and popular algorithms in the library. Among others, we will show how to use out-of-core learning with on-the-fly feature extraction to tackle very large natural language processing tasks, how to exploit an IPython cluster for distributed cross-validation, or how to build and use random forests to explore biological data.

References

[1] L. Buitinck et al. API design for machine learning software: experiences from the scikit-learn project. In *ECML/PKDD Workshop: Languages for Data Mining and Machine Learning*, 2013.

[2] J. D. Hunter. Matplotlib: A 2d graphics environment. *CiSE*, 9: 90, 2007.

[3] T. E. Oliphant. Python for scientific computing. *CiSE*, 9:10, 2007.

[4] F. Pedregosa et al. Scikit-learn: Machine learning in Python. *JMLR*, 12:2825–2830, 2011.

[5] F. Perez and B. E. Granger. IPython: a system for interactive scientific computing. *CiSE*, 9:21, 2007.

[6] S. van der Walt et al. The NumPy array: a structure for efficient numerical computation. *CiSE*, 13:22, 2011.