The virtues and sins of PyMVPA

Yaroslav O. Halchenko[1,2] and Michael Hanke[1,2,3]

[1] *Center for Cognitive Neuroscience, Dartmouth College, Hanover, New Hampshire, USA*
[2] *Department of Psychological and Brain Sciences, Dartmouth College, Hanover, New Hampshire, USA*
[3] *Department of Experimental Psychology, University of Magdeburg, Magdeburg, Germany*

Encouraged by a rise of reciprocal interest between the machine learning and neuroscience communities, multiple studies have demonstrated the superior explanatory power of statistical learning techniques for the analysis of neural data of different acquisition modalities, such as fMRI and EEG. While development and underlying theory of the machine learning methods require substantial skills in computer sciences and statistics, there are fortunately a large number of software packages readily available that implement various promising algorithms. However, these packages are typically implemented in either a very generic fashion or tuned to address a specific problem, with almost none of them geared towards neuroscientific data analysis, thus adoption of those methods is often impaired. To equip community with a specialized platform exposing developments in machine learning to neuroscience, we have developed PyMVPA[1] – Python Multivariate Pattern Analysis[2].

Expressive power of the Python language coupled with a variety of the core libraries for numerical computation in Python or interfaced from other computing platforms (e.g. R through RPy[3]) provide a solid foundation for PyMVPA. Modular design of PyMVPA makes it possible to express complex analysis workflows of neural data in just a few lines of code neither sacrificing readability nor flexibility, while interfacing typical analysis procedures to numerous underlying Python libraries without explicit user awareness of their different API.

To make PyMVPA robust and accessible for neuroscientists, we had to address various demands, from user interface adequacy to the convenience of deployment. This talk will overview the design and capabilities of the framework, and then will accent on particular aspects which could be of an interest to the Python scientific developers: distributions awareness, external dependencies handling, compatibility assurance, data encapsulation, excessive testing, custom logging, and basic literate programming. The talk will conclude with a roadmap of future work planned for PyMVPA development.

[1] M. Hanke, Y. O. Halchenko, P. B. Sederberg, S. J. Hanson, J. V. Haxby, and S. Pollmann. PyMVPA: A Python toolbox for multivariate pattern analysis of fMRI data. Neuroinformatics, 7(1):37–53, Mar. 2009. http://dx.doi.org/10.1007/s12021-008-9041-y

[2] http://www.pymvpa.org

[3] http://rpy.sourceforge.net/rpy2.html