A Python Toolbox for Machine Learning-based Data Analysis

Michael Hanke, Yaroslav O. Halchenko, Per B. Sederberg, Stephen José Hanson, James V. Haxby, Stefan Pollmann

Why Use Machine-learning Based Analysis?
Facilitates multivariate approach to discover distributed activation patterns
Encourages model testing on datasets
Provides direct quantifiable link between experimental conditions and fMRI data
Optimizes the analysis of high-resolution imaging data (Kriegeskorte et al., 2007)
Benefits from inter-disciplinary methodological developments

What Are The Cardinal Features of PyMVPA?
Flexible framework to access existing machine-learning software
Concise sampling interface for human-readable and verifiable code
Modular architecture allowing extensions in multiple languages
Portable code running everywhere from mainframes to cell phones
Data modality independent, but not data modality ignorant
Free and open-source software (FOSS)

Uniform Analysis of Three Different Neuroimaging Data Modalities With Identical Source Code
Sketched out below is a modality-independent cross-validation analysis in PyMVPA (the pre-processing steps are not shown). Various linear classifiers and two featurewise measures are used to compute sensitivity scores. The figures below demonstrate how these sensitivities can be put back into their original context for visualization and interpretation.

# Some classifiers to test
clf = {"SVM": LinearSVC(),
    "CNN": LinearSVMC()
}
# Define some pure sensitivities (or related measures) sensCompAnova = ANOVA(1, oneWayANOVA)
# Store all computed sensitivities
sens = []
# Compute sensitivity, which in turn trains the clf sensitivities = clf.getSensitivityAnalyzer()
# Store sensitivities across splits
sens.append(sens)
# Store across classes
sens.append(sens)

1. fMRI Dataset

2. EEG Dataset

3. Multi-unit Recordings Dataset

Example: Searchlight analysis (Kriegeskorte et al., 2006)

attr = SampleAttributes('sample_attr_filename.txt')
dataset = NiftiDataset(samples=[sub1.bold.nii.gz], labels=attr.labels, chunks=attr.chunks)
cv = CrossValidatedTransferError(TransferError(LinearSVMC()), n_splits=5)
s = Searchlight(cv, radius=1)
s_map = s(dataset)
dataset.save('smap.nii.gz')

Upcoming Features And Future Development
Convenient analysis of event-related datasets
Cluster computing features, solely using FOSS (no licensing costs)
Extended model selection capabilities for automatic tuning of classifier hyperparameters
Graphical User Interface (GUI)

http://www.pymvpa.org