PyMVPA: A Python toolbox for classifier-based data analysis

Michael Hanke1, Yaroslav O. Halchenko2, Per B. Sederberg3, Stephen José Hanson2, James V. Haxby4, Stefan Pollmann1

1) Department of Psychology, University of Magdeburg, Germany
2) Psychology Department, Rutgers Newark, New Jersey, USA
3) Department of Psychology, Princeton University, New Jersey, USA
4) Department of Psychological and Brain Sciences, Dartmouth College, New Hampshire, USA

Classifier-based analysis: Where is the software?

- Multivariate method
- Can focus on distributed patterns instead of activation foci
- Direct quantifiable link between neuroimaging data and experimental manipulation (O’Toole et al., 2007)
- Suitable for high-resolution fMRI imaging (Kriegeskorte et al., 2007)

Advantages of classifier-based analysis:

- Suitable for high-resolution fMRI imaging (Kriegeskorte et al., 2007)
- Suitable for single-trial fMRI (Kriegeskorte et al., 2007)
- Can provide quantitative predictions and classification accuracies

Increasing number of publications applying classifier-based analysis to neuroimaging data (e.g., Haxby et al., 2001; Camitani & Tong, 2005; Hanson et al., 2004; Haynes & Rees, 2007; Hanson & Halchenko, 2008), but only few software packages to facilitate this type of analysis, which are available to a broad audience (Jdsvm, Lecante et al., 2005; MVPA toolbox, Detre et al., 2006).

But: wealth of machine learning software (www.mloss.org)

There is a need for a unifying framework to bridge between established neuroimaging and machine learning software.

PyMVPA Features

- User-centered programmability with a intuitive user interface: Object-oriented toolbox design leading to readable and verifyable code.
- Extensibility: Modular interface to connect extensions in multiple programming languages.
- Transparent reading and writing of datasets: NIfTI support for input and output and additional generic support of various binary and plain text format.
- Portability: Should run on anything supported by Python.
- Open source software: MIT-licensed free software.

Example: Multiple ROI analysis with a Searchlight

```python
attr = SampleAttributes(‘sample_attr_filename.txt’)
dataset = NiftiDataset(
samples=’subj1_bold.nii.gz’,
labels=attr.labels, chunks=attr.chunks,
mask=’subj1roi_mask.nii.gz’)  
cv = CrossValidatedTransferError(
TransferError(LinearCSVMC()), OddEvenSplitter())
si = Searchlight(cv, radius=5)
si_map = si(dataset)
dataset.map2Nifti(array(si_map)).save(’slight.5mm.nii.gz’)
```

Example: Flexible feature selection

```python
attr = SampleAttributes(‘sample_attr_filename.txt’)
dataset = NiftiDataset(
samples=’subj1_bold.nii.gz’,
labels=attr.labels, chunks=attr.chunks)
clf = LinearCSVMC()
clf = FeatureSelectionClassifier( clf, SensitivityBasedFeatureSelector( OneWayANOVA(), FractionTailSelector(0.05, mode=’select’)))
cv = CrossValidatedTransferError(TransferError(clf), NFoldSplitter(), enable_states=[’confusion’])
error = cv(dataset)
```

References


The PyMVPA toolbox is available at http://pkg-exppsy.alioth.debian.org/pymvpa/
PyMVPA mailing list
pkg-exppsy-pymvpa@lists.alioth.debian.org