angelika.lingnau@ur.de 09/2019

Coimbra-Regensburg Advanced Methods in Neuroimaging Workshop

Day 3: RSA Hands-On

Overview

Exercise 1: Basic steps in COSMOMVPA (data set: aka6)
Exercise 2: Visualize behavioural and neural DSMs (data set: aka6)
Exercise 3: Compare behavioural and neural DSMs (data set: aka6)
Multiple regression RSA
Exercise 4: Compare neural DSMs across ROIs (data set: HLR)
Exercise 5: Searchlight RSA (data set: HLR)

Step 1: Getting ready

1. Follow download instructions on http://cosmomvpa.org
[skip if already installed]

2. Save datasets (aka6, HLR) to the following folder:

/…/CoSMoMVPA-Master/data

3. Save exercises to the following folder:

/…/CoSMoMVPA-Master/handsOnCoimbra

Step 2: Have a look at the format of the CoSMoMVPA dataset:

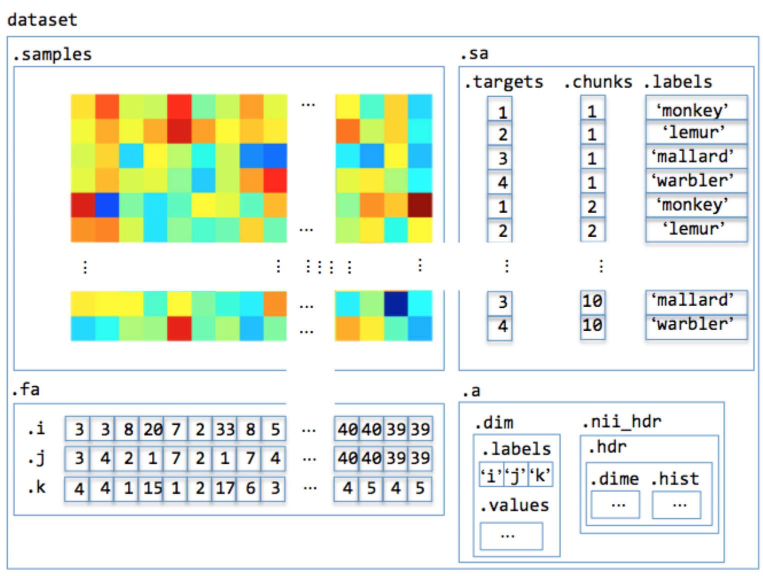

Figure 1: Format of CoSMoMVPA dataset.

Step 3: Description of the two data sets used in the RSA Hands-On sessions

Data set 1: aka6

Based on Connolly et al (2012), Representation of biological classes in the human brain. Journal of Neuroscience, doi 10.1523/JNEUROSCI.5547-11.2012

*Six categories (monkey, lemur, mallard, warbler, ladybug, lunamoth) during ten runs in an fMRI study. Using the General Linear Model responses were estimated for each category in each run, resulting in 6*10=60 t-values for each voxel.*

Data set 2: High level retinotopy (HLR)

Based on Pfannerstill & Lingnau (in prep.)

Visual stimulation of the four quadrants (upper left, upper right, lower left, lower right) during two runs in an fMRI study. Using the General Linear Model, responses were estimated for each quadrant in each run and averaged across the two runs, resulting in 4 t-values for each voxel.

Step 4: Starting the exercises

Exercise 1. Getting familiar with CoSMoMVPA.

In Matlab, open the first exercise (exercise1.m):

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%% EXERCISE I: GETTING FAMILIAR WITH COSMOMVPA %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% angelika.lingnau@ur.de, 09/2019

%(1) Load dataset after specifying targets
pathToData = '/Users/alingnau/CoSMoMVPA-master/data/ak6/';
subjectID = 's01';
fName = 'glm_T_stats_perrun.nii';
ds = cosmo_fmri_dataset([pathToData, subjectID, '/', fName]);

%have a look at the contents of ds
ds
cosmo_disp(ds)
```

Adjust the path in 'pathToData'.

This first example shows how to load a data set with CoSMoMVPA, using the function cosmo_fmri_dataset [note: this function works for a range of different inputs, including AFNI, SPM, BV, ANALYZE, and NIFTI0. Inspect help cosmo_fmri_dataset for further details].

Run the code of the first section in exercise_1 and inspect the output on the screen. What is the size of ds.samples?

The second section of example1 shows how to load a data set with CoSMoMVPA after explicitly specifying the targets (see Figure 1 above). Run the code and inspect the output on the screen again. Notice the difference.

```
%(2) Load dataset after specifying targets
targets=repmat(1:6,1,10)';
ds = cosmo_fmri_dataset([pathToData, subjectID, '/', fName], ...
    'targets', targets);

%have a look at the contents of ds
ds
cosmo_disp(ds)
```

The third section of example1 shows how to select samples from a data set, using the function cosmo_slice (see help cosmo_slice for further details). In this example, the first six samples (corresponding to conditions 1:6, run 1) are selected.

```
%(3) Slice dataset
targetsToSelect = (1:1:6)'
ds_sliced = cosmo_slice(ds, targetsToSelect);

%have a look at the contents of ds_sliced
ds_sliced
cosmo_disp(ds_sliced)
```

The fourth section shows to plot individual slices using the function cosmo_plot_slices. Using this function, try to plot slices corresponding to conditions 1:6 of the first run in six separate plots.

```
%(4) Show some slices
%show slices of first condition, first run
cosmo_plot_slices(cosmo_slice(ds, 1));

%your own code here:
%in six separate plots, show conditions 1:6 of the first run

%XXXXXX
```

The next section shows how to load a data set from a region of interest (in this example: early visual cortex). Try out the code and have a look at the content of ds_masked.

```
%(5) Load dataset after specifying targets and a mask
maskName = 'ev_mask.nii';

ds_masked = cosmo_fmri_dataset([pathToData, subjectID, '/', fName], ...
    'mask', [pathToData, subjectID, '/', maskName],...
    'targets', targets);
```

Using your own code, create two masked data sets ds_masked_odd and ds_masked_even from early visual cortex that contain the mean across odd and even runs, separately for each of the six condition

```
%(7) your own code here:
%create two masked data sets ds_masked_odd and ds_masked_even that contain
%the mean across odd and even runs, separately for each
%of the six conditions
```

Exercise 2. Visualizing behavioural and neural dissimilarity matrices (DSMs).

This exercise shows how to load and visualize behavioural and neural DSMs. Have a look at the code to understand the different functions that are used. Run the code and inspect the resulting images.

Exercise 3. Comparing behavioural and neural DSMs.

This exercise shows how to compare behavioural and neural DSMs using the function cosmo_target_dsm_corr_measure. Before running the code, have a look at help cosmo_target_dsm_corr_measure for further details. *Note that it is recommended to set the argument 'center_data' to true when using the default 'correlation' metric. If set to true, the mean of each feature (column in ds.samples) is subtracted from each column prior to computing the pairwise distances for all samples in ds. The rationale behind this recommendation can be found in Diedrichsen & Kriegeskorte (2017). Representational models: A common framework for understanding encoding, pattern-component, and representational-similarity analysis. PLoS Computational Biology, 13(4), e1005508*

After running the code, create your own code to run the same analysis again, with the argument 'center_data' set to 'true'. Inspect the difference.

For advanced users: set up the analysis as a multiple regression RSA, using both behavioural models. Do this once with uncentered and once with centered data. Inspect the differences. Hint: use the argument glm_dsm for the function cosmo_target_dsm_corr_measure.

Exercise 4. Comparing neural DSMs across ROIs.

This exercise shows how to compare neural DSMs across ROIs using the function cosmo_target_dsm_corr_measure. Run the code and inspect the resulting figures.

Exercise 5. RSA Searchlight.

This exercise shows how to compute an RSA searchlight using the function cosmo_searchlight. Before running the code, inspect help cosmo_searchlight for details.

Your own code: compute an RSA searchlight on the aka6 dataset, using the model 'behav'.