Dataset with brain atlas A for classification tasks: for first set of 5 small programming tasks: Oasis dataset provided directly by the *nilearn* package

Can be retrieved by downloading *nilearn.datasets.fetch\_oasis\_vbm(n\_subjects=100)*, roughly 900 MB, structural brain scans (so-called voxel-based morphometry) **for** **classification** of male and female individuals

*import numpy as np*

*from nilearn import datasets*

*from nilearn.input\_data import NiftiLabelsMasker*

*from nilearn.image import index\_img*

*import nibabel as nib*

*brain\_data = datasets.fetch\_oasis\_vbm(n\_subjects=100)*

*yeo = datasets.fetch\_atlas\_schaefer\_2018(n\_rois=100) # this needs to be modified for some tasks*

*print('%i regions found in this atlas' % (len(np.unique(nib.load(yeo.maps).get\_data().ravel())) - 1))*

*masker = NiftiLabelsMasker(labels\_img=yeo.maps, standardize=True,*

*memory='nilearn\_cache')*

*input\_variables = masker.fit\_transform(brain\_data.gray\_matter\_maps)*

*output\_variable = np.array(brain\_data.ext\_vars.mf == b'F', dtype=np.int) # gives 1 for females and 0 for males*

Task 1- Use the PyMC3 package (<https://docs.pymc.io>) to implement a Bayesian hierarchical Logistic Regression (tip: use PyMC3.Bernoulli() and PyMC3.invlogit()) to classify sex. Each bottom-level region slope variable (zscored) should be modeled as a Gaussian distribution priors (mu=0, sd=1). All subject data is fitted in the same quantitative model, that is, all datapoints are used at once. As binary higher hierarchical level please implement a hyperprior based on the variance component of the lower-level region slopes by pooling through a higher-level HalfCauchy distribution for their respective lower-level region slopes. As binary hyper-prior category please use left versus right brain side of the 100 schaefer-yeo atlas regions. 500 MCMC iterations is enough. Report output of PyMC3.summary(credible\_interval=0.90).

Task 2- Same as task 1, but using a higher-level Gaussian for the mu-parameter (not the variance component) as a hyper-prior for the lower-level region slopes. 500 MCMC iterations is enough. Report output of PyMC3.summary(credible\_interval=0.90).

Task 3- Use the PyMC3 package (<https://docs.pymc.io>) to implement a Bayesian hierarchical Logistic Regression (tip: use PyMC3.Bernoulli() and PyMC3.invlogit()) to classify sex. Each bottom-level region slope variable (zscored) should be modeled as a Gaussian distribution (mu=0, sd=1). All subject data is fitted in the same quantitative model. As binary hyperprior please implement a hyperpriors based on the variance component of the lower-level region slopes by pooling through a higher-level HalfCauchy distribution for their respective lower-level region slopes. As binary hyper-prior category please use high versus low age (median-split to find cut-off) of the 100 schaefer-yeo atlas regions. 500 MCMC iterations is enough. Report output of PyMC3.summary(credible\_interval=0.90).

Task 4- Same as task 3, but using a higher-level Gaussian for the mu-parameter (not the variance component) as a hyper-prior for the lower-level region slopes. 500 MCMC iterations is enough. Report output of PyMC3.summary(credible\_interval=0.90).

Task 5- Same as 3, but creating 2 higher-level categories based on median-split of the ‘delay’ variable supplied by the OASIS dataset. 500 MCMC iterations is enough. Report output of PyMC3.summary(credible\_interval=0.90).

Dataset with brain atlas B for continuous prediction tasks: for second set of 5 small programming tasks: Oasis dataset provided directly by the *nilearn* package

Can be retrieved by downloading *nilearn.datasets.fetch\_oasis\_vbm(n\_subjects=100)*, roughly 900 MB, structural brain scans (so-called voxel-based morphometry) to predict of **continuous outcome** givena subject’s age from structural brain scans

*import numpy as np*

*from nilearn import datasets*

*from nilearn.input\_data import NiftiLabelsMasker*

*from nilearn.image import index\_img*

*from sklearn.preprocessing import StandardScaler*

*import nibabel as nib*

*brain\_data = datasets.fetch\_oasis\_vbm(n\_subjects=100)*

*crad = datasets.fetch\_atlas\_craddock\_2012()*

*atlas\_nii = index\_img(crad['scorr\_mean'], 42) # this needs to be modified for some tasks*

*print('%i regions found in this atlas' % (len(np.unique(atlas\_nii.get\_data().ravel())) - 1))*

*masker = NiftiLabelsMasker(labels\_img=* atlas\_nii*, standardize=True)*

*input\_variables = masker.fit\_transform(brain\_data.gray\_matter\_maps)*

*output\_variable = StandardScaler().fit\_transform(brain\_data.ext\_vars.age[:, None])[:, 0] # gives subject age on standard units after z-scoring*

6- analogous to task 1, but implementing a Bayesian hierarchical regression (with continuous outcome variable) to predict subject age (zscored)*.*

7- analogous to task 2, but implementing a Bayesian hierarchical regression (with continuous outcome variable) to predict subject age (zscored)*.*

8- analogous to task 3, but implementing a Bayesian hierarchical regression (with continuous outcome variable) with sex as binary higher level to predict subject age (zscored)*.*

9- analogous to task 4, but implementing a Bayesian hierarchical regression (with continuous outcome variable) with sex as binary higher level to predict subject age (zscored)*.*

10- analogous to task 5, but implementing a Bayesian hierarchical regression (with continuous outcome variable) to predict subject age (zscored)*.*