

Using stSpikeSlabEEG

Shariq Mohammed

R package for “[Mohammed, S.](#), Dey, D.K. and Zhang, Y., 2020. Classification of high-dimensional electroencephalography data with location selection using structured spike-and-slab prior. Statistical Analysis and Data Mining: The ASA Data Science Journal, pp.1-17. <https://doi.org/10.1002/sam.11477>”

Code to run the model in (Mohammed et.al, 2020) on the EEG data using the package stSpikeSlabEEG.

Contents

1. Load EEG data and divide it into training and test.
2. Build the model proposed in the manuscript with the training data and predict for the test data set.
3. Run a 5-fold cross-validation with the EEG data provided in the package.

```
# install the package (devtools package needed)
if(!require(devtools)) install.packages("devtools")
devtools::install_github('shariq-mohammed/stSpikeSlabEEG')

# Load the package
library(stSpikeSlabEEG)
```

Load EEG data and divide it into training and test

Load EEG Data

```
X = stSpikeSlabEEG::X_eeg
```

Load location indices for which distance matrix is available

```
ind64to57 = stSpikeSlabEEG::ind64to57_eeg
```

Update EEG data for locations above

```
X = X[,ind64to57,]
```

Load binary response

```
y = t(stSpikeSlabEEG::y_eeg)
```

Dimensions of EEG data

```
tau = dim(X)[1]
p = dim(X)[2]
n = dim(X)[3]
```

Scale data at subject level by its Frobenius norm

```
X_int = sapply(1:n, function(i) sqrt(sum(X[,i]^2)))  
X_sc = array(NA, dim = dim(X))  
for(i in 1:n) X_sc[,i] = X[,i]/X_int[i]
```

Sample indices for training data

```
trn.ind = sample.int(n, size = 100)
```

Responses for training and testing

```
y.trn = y[trn.ind] # responses from training data  
y.test = y[-trn.ind] # responses from test data
```

EEG data for training and testing

```
X.trn = aperm(X_sc[,trn.ind], c(3,2,1)) # training EEG data  
X.test = aperm(X_sc[, -trn.ind], c(3,2,1)) # test EEG data
```

Build the model proposed in the manuscript with the training data and predict for the test data set

Fit the model

```
modelFit = strucBayes(y = y.trn, X = X.trn, dist.mat = dist.mat,  
                      Nmcmc = 1000, burnin = 100, thin = 10, # MCMC settings  
                      X.test = X.test)
```

Estimates for the coefficients, local predicted probabilities.

```
modelFit$b.strucBayes  
modelFit$p.strucBayes.pred
```

Response prediction for the test data set

```
modelFit$y.strucBayes.pred
```

Run a 5-fold cross-validation with the EEG data provided in the package

```
cvResults = crossVal(seed = 113, # seed for creating fold-splits  
                    Nmcmc = 1000, burnin = 100, thin = 10)
```