brainSimulator Documentation

Release 0.5.1

SiPBA@UGR

Nov 13, 2017

CONTENTS:

Functional brain image synthesis using the KDE or MVN distribution. Currently in beta. Python code.

brainSimulator is a brain image synthesis procedure intended to generate a new image set that share characteristics with an original one. The system focuses on nuclear imaging modalities such as PET or SPECT brain images. It analyses the dataset by applying PCA to the original dataset, and then model the distribution of samples in the projected eigenbrain space using a Probability Density Function (PDF) estimator. Once the model has been built, anyone can generate new coordinates on the eigenbrain space belonging to the same class, which can be then projected back to the image space. Created on Thu Apr 28 15:53:15 2016 Last update: 9 Aug, 2017

@author: Francisco J. Martinez-Murcia <fjesusmartinez@ugr.es>

class brainSimulator.**KDEestimator**(*bandwidth=1.0*)

An interface for generating random numbers according to a given Kernel Density Estimation (KDE) parametrization based on the data.

botev_bandwidth(*data*)

Implementation of the KDE bandwidth selection method outline in:

26. (a) Botev, J. F. Grotowski, and D. P. Kroese. *Kernel density estimation via diffusion.* The Annals of Statistics, 38(5):2916-2957, 2010.

Based on the implementation of Daniel B. Smith, PhD. The object is a callable returning the bandwidth for a 1D kernel.

Forked from the package *PyQT_fit <https://code.google.com/archive/p/pyqt-fit/>*.

Parameters **data** (*numpy.ndarray*) – 1D array containing the data to model with a 1D KDE.

Returns Optimal bandwidth according to the data.

finite(*val*)

Checks if a value is finite or not

class brainSimulator.**BrainSimulator**(*method='kde'*, *algorithm='PCA'*, *N=100*, *n_comp=-1*, *regularize=False*, *verbose=False*)

createNewBrains(*N*, *kernel*, *components=None*)

Creates new samples from the model.

generateDataset(*stack*, *labels*, *N=100*, *classes=None*, *components=None*)

Fits the model and generates a new set of N elements for each class specified in "classes".

Parameters

- **stack** (*numpy.ndarray*) – the stack from which the model will be created
- **labels** (*numpy.ndarray*) – a vector containing the labels of the stacked dataset

- **N** (either int (the same N will be generated per class) or a list of the same length as *classes* containing the number of subjects to be generated for each class respectively.) – the number of elements (per class) to be generated
- **classes** (a list of the classes to be generated, e.g.: *[0, 2]* or *['AD', 'CTL']*.) – the classes that we aim to generate
- **components** (*integer*) – the number of components used in the synthesis. This parameter is only valid if *components* here is smaller than the *n_comp* specified when creating and fitting the ‘BrainSimulator‘object.

Returns array with labels of the synthetic stack

CHAPTER

ONE

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

b

INDEX