GaussBetti

Release 1.0.0

Siddharth

Jul 19, 2021

CONTENTS

TOPOLOGICALFUNC

topologicalFunc.**GaussianFiltration**(*GaussianRandomField*, *type='lower'*)
    Generates Filtration for the Gaussian Random Field.

> **Parameters**
>
> - **GaussianRandomField** (`array`) – numpy 2-D array. The Gaussian Random Field generated from the class using Gen_GRF method.
> - **type** (`string`) – Takes ipnut either 'lower' or 'upper' for lower or upper filtration.
> - **nsize** (`integer`) – Size of the Gaussian Random Fields grid.
>
> **Returns** Filtration Diagram
>
> **Return type** Dionysus object

topologicalFunc.**GenerateBettiP**(*Filtraion*, *thresholds_start*, *thresholds_stop*, *type='lower'*)
    Generates the Betti numbers from the Filtration diagram.

> **Parameters**
>
> - **Filtration** (`Dionysus object`) – Output of GaussianFiltration.
> - **thresholds_start** (`float`) – start value for generating superlevels of the Gaussian Random field .
> - **thresholds_stop** (`float`) – stop value for generating superlevels of the Gaussian Random field.
>
> **Returns** Multidimensaion array contaiing Betti numbers for different dimensions
>
> **Return type** Numpy array

topologicalFunc.**GenerateGenus**(*Betti_array*)
    Generates the Genus curve for gaussian random field using Betti arrays.

> **Parameters** `array` (`Betti`) – Betti array from GenerateBettiP.
>
> **Returns** 1-D array contaiing Genus curve for the Gaussian random field.
>
> **Return type** Numpy array

UTILITIES

utilities.**KLdivergence**(*x*, *y1*, *y2*)

> Calculates the KL divergence for 2 different Gaussian Random Field.

> **Parameters**
>
> > - **x** (`array`) –
> >
> > - **y1** (`array`) – Gausian Random Field of null hypothesis as a 1-D array
> >
> > - **y2** (`array`) – Gaussian Random Field of test hypothesis as a 1-D array
>
> **Returns** KL divergence
>
> **Return type** float

utilities.**SaveROC**(*PFA*, *PD*, *nsize*, *num_iter*, *H0*, *H1*, *Betti*, *type*)

> Saves the PFA and PD array with the labels provided through parameters.

> **Parameters**
>
> > - **PFA** (`array`) – numpy vector. The PFA array generated during ROC gen.
> >
> > - **PD** (`array`) – numpy vector. THe PD array generated during ROC gen.
> >
> > - **nsize** (`integer`) – Size of the Gaussian Random Fields grid.
> >
> > - **num_iter** (`integer`) – Number of iteration for which ROC gen is run.
> >
> > - **H0** (`integer`) – Power spectral index of Null Hypothesis.
> >
> > - **H1** (`integer`) – Power spectral index of Test Hypothesis.
> >
> > - **type** (`string`) – type of the ROC curve generated
>
> **Returns** None
>
> **Return type** None

utilities.**likelihoodratio**(*correlation0*, *correlation1*, *X0*, *X1*)

> Calculates the likelihood ratio for the False alarm and detection.

> **Parameters**
>
> > - **correlation0** (`array`) – Correlation matrix for the Gaussian Random Field for null hypothesis.
> >
> > - **correlation1** (`array`) – Correlation matrix for the Gaussian Random Field for test hypothesis.
> >
> > - **X0** (`array`) – Gausian Random Field of null hypothesis as a 1-D array
> >
> > - **X1** (`array`) – Gaussian Random Field of test hypothesis as a 1-D array

Returns likelihood ratio

Return type float

utilities.plotROC(*PFA*, *PD*, *nsize*, *num_iter*, *H0*, *H1*, *Betti*, *type*)
    Plots the PFA and PD ROC graph with the labels provided through parameters.

Parameters

- **PFA** (*array*) – numpy vector. The PFA array generated during ROC gen.

- **PD** (*array*) – numpy vector. THe PD array generated during ROC gen.

- **nsize** (*integer*) – Size of the Gaussian Random Fields grid.

- **num_iter** (*integer*) – Number of iteration for which ROC gen is run.

- **H0** (*integer*) – Power spectral index of Null Hypothesis.

- **H1** (*integer*) – Power spectral index of Test Hypothesis.

- **type** (*string*) – type of the ROC curve generated

Returns None

Return type None

GAUSSCLASS

class gaussClass.`GaussianRandomField`(*Nsize, n*)

The class for making Gaussian random field with specified spectral index and size of grid.

Nzise

size of the grid.

> **Type** int

n

Spectral index of the power law used to generate the Gaussian Random Field.

> **Type** int

k_ind

Grid in the fourier space.

> **Type** array

PowerSpectrum

The power spectrum grid made using the spectral index used to make the Gaussian Random Field.

> **Type** array

corr_s

Correlation matrix in the fourier space.

> **Type** array

corr_f

Correlation matrix in the spatial space.

> **Type** array

Gen_GRF(*type='grid'*)

GenerateBettiP

Generates the Gaussian Random field with the specified paramters.

> **Parameters** `type` (`str`) – Takes either 'grid' or 'array' in string format

Returns: Numpy array: Gaussian Random field

PowerSpectrum_grid_generator()

Generates the powerspectrum grid.

fourier_space_ind()

Generates the fourier space grid.

gen_correlation()

Generates the correlation matrices in fourier and spatial spcae.

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

INDEX