
sampen Documentation

Release 0.0.16

Joe Bergantine

Jan 04, 2020

Contents:

1	Installation	3
1.1	Via PIP	3
1.2	Via Git	3
2	Usage	5
3	Contributing	7
3.1	Running Tests	7
3.2	Documentation	7
3.3	Tagging Releases	8
3.4	Pushing Releases to PyPi	8

Partial port of v1.2 C-language code for estimating Sample Entropy using SampEn from PhysioNet to Python.

C-language code last updated 1 November 2004, by George Moody.

Original author Doug Lake. dlake@virginia.edu

CHAPTER 1

Installation

1.1 Via PIP

```
$ pip install sampen
```

1.2 Via Git

Clone the repo:

```
$ git clone git@github.com:bergantine/sampen.git
```

This will create a new directory with the repo in it. Change into that directory and install:

```
$ cd sampen && python setup.py install
```


CHAPTER 2

Usage

Unlike the original C code which loads time series data either from standard input or a file, this script expects the time series to be expressed in a list of floats. If the time series exists in a flat text file that can be loaded into Python and saved into a list::

```
from sampen import sampen2

# initialize a list
series_data = []

# open the file and read each line into the list
with open('relative/path/to/file.txt', 'r') as file:
    for row in file:
        series_data.append(float(row.strip(' \t\n\r')))

# calculate the sample entropy
sampen_of_series = sampen2(series_data)
```

The default maximum epoch length (m) is 2, the default tolerance (r) is 0.2.

The estimate of the conditional probability that the subseries of the epoch length that matches pointwise within the tolerance (that is, the Sample Entropy) will be the last item in the returned tuple.

The earlier items in the returned tuple will be the sample entropies for lengths 0 up to the maximum epoch.

Therefore, inspecting the returned data:

```
>>> sampen_of_series
```

Returns something like::

```
[
  (0, 2.140629540027156, 0.0028357991885715863)
  (1, 2.162868347337613, 0.004903248034526253),
  (
```

(continues on next page)

(continued from previous page)

```
    # Epoch length for max epoch
    2,
    # SampEn
    2.123328492035711,
    # Standard Deviation
    0.007596323621379352
),
]
```

1. Fork the repository on GitHub
2. Create a named feature branch (like *feature/add_component_x*)
3. Write your change
4. Write tests for your change (if applicable)
5. Run the tests, ensuring they all pass
6. Submit a Pull Request using GitHub

3.1 Running Tests

```
$ python setup.py test
```

3.1.1 With TOX

First, install Tox, then run the tests.

```
$ pip install tox
$ tox
```

3.2 Documentation

Documentation is written in [ReStructuredText](#) and built with [Sphinx](#).

1. Install `sphinx` and `sphinx-autobuild` as necessary via `pip`
2. Edit `docs/index.rst`

3. Build the HTML with `$ make html` from within the `docs` directory

3.3 Tagging Releases

Update `version` and `download_url` in `setup.py`.

Update the `version` and `release` in `docs/conf.rst`

Remake documentation

3.4 Pushing Releases to PyPi

Install `wheel` and `twine` as necessary via `pip`

Where `XX` is the version (e.g. `0.0.15`)

```
python setup.py sdist
python setup.py bdist_wheel --universal
twine upload dist/sampen-XX* -r pypitest
twine upload dist/sampen-XX*
```