DPApipeline Documentation

Release 0.1

Maria d'Errico

January 31, 2020

CHAPTER 1

Introduction

1.1 Density Peaks Advanced clustering

The DPApipeline package implements the Density Peaks Advanced clustering algorithm as introduced in the paper *Automatic topography of high-dimensional data sets by non-parametric Density Peak clustering* [1]. The package offers the following features:

- Intrinsic dimensionality estimation by means of the TWO-NN algorithm
- Adaptive k-NN Density estimation by means of the PAk algorithm
- Advanced version of the DP clustering algorithm, including an automatic search of cluster centers and assessment of statistical significance of the clusters

The top-level directory layout:

```
cd DPApipeline
ls -l
```

```
.
|-- data/                              # Input and output files.
|-- docs/                              # Documentation files.
|-- notebooks/                         # Python scripts in Jupyter notebooks.
|-- Pipeline/                          # Source files.
|-- README.rst
|-- requirements.txt
|-- run_ipynb2py_versioning.sh
|-- config.sh
|-- setup.py
```

1.1.1 Source files

The source Python codes are stored inside the `Pipeline` folder:

```
cd Pipeline
ls -l
```

```
.
|-- ...
|-- Pipeline/
|   |-- __init__.py
|    |-- DPA.py           # Python module implementing the DPA
|    |                    # clustering algorithm.
|    |
|    |-- _DPA.pyx         # Cython extension of the DPA module.
|    |
|    |-- PAk.py           # Python module implementing the PAk
|    |                    # density estimator.
|    |
|    |-- _PAk.pyx         # Cython extension of the PAk module.
|    |
|    |-- NRmaxL.f90       # Fortran extension for the Newton-Rapson algorithm.
|    |
|    |-- twoNN.py         # Python module implementing the TWO-NN
|                         # algorithm for the ID calculation.
|
|-- ...
```

1.1.2 Documentation files

Full documentation about the Python codes developed and the how-to instructions is crested in the `doc` folder using *Sphinx*. The `DPApipeline.pdf` is in the `doc/_build/rinioh` folder.

1.1.3 Jupyter notebooks

Examples of how-to run the `DPA`, ``PAk` and ``twoNN` modules are provided as Jupyter notebooks in the `notebooks` folder. Additional useful user-cases are available in the same folder.

```
.
|-- ...
|-- notebooks/
|    |-- DPA_analysis.ipynb                 # Guided example of how-to run the
Pipeline package.
|
|
|-- ...
```

1.2 Getting started

The source code of DPApipeline is on github DPApipeline repository.

You need the `git` command in order to be able to clone it, and we suggest you to use Python virtual environment in order to create a controlled environment in which you can install DPApipeline as normal user avoiding conflicts with system files or Python libraries.

The following section documents the steps required to install DPApipeline on a Linux or Windows/Mac computer.

1.2.1 Debian/Ubuntu

Run the following commands to create and activate a Python virtual environment with *python virtualenv*:

```
apt-get install git python-dev virtualenv*
virtualenv -p python3 venvdpa
. venvdpa/bin/activate
```

1.2.2 Windows

A possible setup makes use of Anaconda. It has preinstalled and configured packages for data analysis and it is available on all major platforms. It uses *conda* as package manager, in addition to the standard pip.

A versioning control can be installed by downloading git.

Run the following commands to activate the conda virtual environment:

```
conda create -n venvdpa
conda activate venvdpa
```

to list the available environments you can type `conda info --envs`, and to deactivate an active environment use `source deactivate`.

1.3 Installation

Assuming you already have the Python virtual enviroment installed and activated on your machine, run the following commands to download the DPApipeline source code:

```
git clone https://airamd@bitbucket.org/airamd/dpapipeline.git
```

Install DPApipeline with the following commands:

```
cd dpapipeline
. compile.sh
```

Note that it is possible to check which packages are installed with the `pip freeze` command.

1.3.1 Quickstart

A use-case example is provided in the DPA_analysis.ipynb jupyter notebook.

- *Index*
- *Module Index*
- *Search Page*