Installation and Requirements

Patricia Wollstadt edited this page on Jan 15, 2024 · 36 revisions

Download or clone the latest version from GitHub, unpack and run (from the folder containing IDTxl's setup.py file)

```
pip install .
```

or

```
pip install -e .
```

for 'editable' mode.

We recommend the installation in a conda environment.

Pages 49

Wiki Home

Full documentation

Theoretical background

Tutorials

Clone this wiki locally

https://github.com/pwollstadt/IDTxl.

Requirements

- python 3.4.3+
- numpy
- scipy
- cffi
- h5py

IDTxl provides both CPU- and GPU-accelerated estimators, you have to install the requirements for at least one of the two setups. If you plan on using the GPU estimators, install:

- pyopencl 2015.1 (see this installation guide, it is recommended to install pyopencl via your distribution's package manager or from within an anaconda environment as described here)

If you plan on using the CPU estimators, install:

- JPype1 0.7+ (requires `g++` , and `python-dev`), latest version was tested with JPype 1.2.1

- java jdk 1.6+ (e.g. openjdk)
- libffi-dev

If you want to use IDTxl's plotting routines, install:

- networkx
- matplotlib

If you want to use the Tartu PID estimator, install:

- ecos
- the platform has to support `numpy.float128`

If you want to use the Goettingen, shared-exclusion PID estimator, install:

- PrettyTable

If you want to use the Rudelt history-dependence estimator (HDE) for spike timing data, install:

- mpmath
- cython
- and run the setup script as described in the installation instructions

If you want to use the Numba CUDA estimator, an appropriate NVIDIA Driver see installation guide and NVIDIA CUDA Toolkit see installation guide for the used NVIDIA graphic card needs to be installed on the system. (WARNING: numba 0.52 does not support cuda 11.1 or higher!) Then install the following python packages:

- numba (see installation guide)
- cudatoolkit

Note: Newer versions of JPype1 will most likely work, they just haven't been tested yet. Don't confuse jpype1 (which you'll need for IDTxl) with either jpype, or jpype1-py3. If you installed pyopencl from within an anaconda environment, it may happen that pyopencl doesn't find the OpenCL driver (`pyopencl.cffi_cl.LogicError: clGetPlatformIDs failed: <unknown error -1001>`). In this case, you have to copy the *.icd files from /etc/OpenCL/vendors to [PATH_TO_ANACONDA]/envs/[ENVIRONMENTNAME]/etc/OpenCL/vendors, see here.

Anaconda installation

A full installation in Anaconda for use with both GPU- and CPU-backends may look like this

```
git clone https://github.com/pwollstadt/IDTxl.git
conda create --name idtxl python=3 pip matplotlib h5py scipy networkx
```

```
conda activate idtxl
conda install -c conda-forge jpype1  # required by CPU JIDT estimators
conda install -c conda-forge pyopencl  # required by GPU OpenCL estimators
conda install -c anaconda ecos  # required by Tartu PID estimator
conda install numba # required by NumbaCuda estimators
conda install cudatoolkit # required by NumbaCuda estimators

cd IDTxl
pip install -e .
python demos/demo_bivariate_mi.py
```

Not that installation of individual packages is optional if you don't intend to use the respective estimators.

Step-by-step installation of Python packages on Ubuntu

IDTxl has been tested on the following Ubuntu releases:

- 14.04 (java)
- 15.10 (java, opencl)
- 15.04
- 16.04

Installing the required python packages can be done via Ubuntu's package management program apt-get. It should be as easy as running:

```
sudo apt-get install python3-numpy python3-scipy python3-cffi python3-h5py python3-networkx python3-matplotlib

#GPU
sudo apt-get install python3-pyopencl

#CPU
sudo apt-get install g++
sudo apt-get install python3-dev
sudo apt-get install python3-jpype1 #if it fails, try sudo pip3 install jpype1
```

Windows users

Good luck! :)

(However, installation in a conda environment works under Windows in most cases!)

IDTxl-Wiki Home