

Getting Started

Introduction

The Gaussian Process Summer School will include some hands-on tutorials in which we will build some simple Gaussian process models. The tutorials will be in Python, featuring the open source *GPy package* that has been developed by the Machine Learning group at the University of Sheffield.

Please bring your own laptop. Prior Python programming skills are not required, however you should ensure that you have installed the appropriate version of Python and the packages/libraries we will be using:

- **Python 3.5** (or a later version)
- `numpy`
- `scipy`
- `matplotlib`
- `GPy`
- `jupyter`

We highly recommend that you install an integrated Python environment, in particular [Anaconda](#) which will allow for easy installation of packages. It also comes with the latest versions of `numpy` and `scipy`.

All labs are in a format called "*notebooks*", which can be run using [Jupyter](#). These are worksheets that can execute Python code in blocks in your web browser.

The following instructions will tell you how to install and setup the Python library for the tutorials, and some information on installing and running Jupyter.

Installing Python with Anaconda

Anaconda is a distribution of the Python programming language that comes integrated with a number of precompiled libraries, and its own package and environment manager, called `conda`. It freely allows use of installation of packages and libraries via `conda` or `pip`. We recommend using Anaconda to manage your Python language environment, *particularly if you are new to Python*, and the following instructions will assume you are using Anaconda. If you are using a different Python distribution, you may have to tailor to following instructions, but you should ensure that you are using Python 3.5+.

Installing

The easiest way to get a working Python environment is to install Anaconda. It is fairly straightforward to install, but can take some time so you must make sure this is done before the lab.

1. Download and install the free version of Anaconda from its webpage:
<https://www.anaconda.com/download>, selecting the *Python 3.6 version* appropriate for your operating system
 - Windows: the installer will be a `.exe` executable, and you can follow the setup as instructed
 - Linux: the installer is a `.sh` shell script, and you can run it in the terminal and follow the setup as instructed
 - Note you may have to enable execution of the file, by either
 - Right click the file and select Properties, and under `Permissions` check "Allow executing file as program"
 - `$ chmod +x /path/to/installationfile.sh`
 - macOS: the installer is a `.pkg` software package, and you can follow the setup as instructed
2. Update Anaconda, `numpy`, `scipy`, and `matplotlib`: open a command prompt or terminal and execute the following commands
 - i. `conda update -y anaconda`
 - ii. `conda update -y numpy scipy matplotlib`
3. Update `jupyter`
 - i. `conda update -y jupyter`
 - ii. If you are not using Anaconda, you can install `jupyter` by calling `$ python3 -m pip install jupyter`

Installing GPy

Important: (advanced) before installing `GPy`, you may want to create a special Python environment for the summer school. Complete the steps to create a new environment (below) before performing the following. See [Advanced: Creating a Python Environment for the Labs](#) for details.

We will install `GPy` using `pip`, by performing the following command in the terminal/command prompt:

```
$ pip install GPy
```

Installing `GPy` will also install its other dependencies, such as `paramz`, so you don't need to worry about these. You can test that the installation of `GPy` is working by running the following in a Python shell:

```
>>> import GPy
>>> GPy.tests()
```

This should run through the testing sequence.

Running the Lab Sheets

On the day of each tutorial, the respective lab sheet will be made available to download here:

- Day 1: Gaussian Process Regression [download lab sheet \(.ipynb\)](#)
- Day 2: GPs for Non-Gaussian Likelihoods and Big Data [download lab sheet \(.ipynb\)](#)
- Day 3: Bayesian Optimisation with Gaussian Processes [download lab sheet \(.ipynb\)](#)

To use these notebooks, you should download the respective lab sheet. In a terminal window, navigate to the files path (using the `cd` command) and run the command `jupyter notebook`. This will open a browser window connecting to the (locally hosted) notebook server. The notebook should be visible in the file list.

Typically, Jupyter will launch the server at <http://localhost:8888/tree>, and you can navigate to it this way, should you accidentally close the window.

Advanced: Creating a Python Environment for the Labs

It may be preferable to create a custom 'virtual' Python environment for use in the labs. A virtual environment gives you an isolated working copy of Python with its own files and packages, allowing you to specify versions of the Python executable and packages without affecting your main distribution. For example, if you predominantly use Python 2, you may want to create a Python 3 environment separately for using the lab sheets (which make use of functionality not available in Python 2).

The following instructions assume you are using **Anaconda**, but if not you may be able to replicate the behaviour using `virtualenv`.

Creating a New Environment with Conda

Full details of the functionality of `conda`'s virtual environment are available on its [documentation page](#). The following instructions should get you a working environment with all the necessary features using a terminal. It is also possible to use Anaconda Navigator, if you have it installed, to perform these steps using a graphical user interface.

In the following steps, we will name our environment " `python_3_gpss` ", but you can name it whatever is most appropriate (just make sure you replace any mention of `python_3_gpss` with your preferred name). Likewise, we will use Python 3.6, though 3.5 may also be used, in which case replace `3.6` with `3.5` as appropriate. Setting the version is particularly important if you use Python 2 with Anaconda by default.

In a command prompt / terminal, execute the following line:

```
$ conda create -y --name python_3_gpss python=3.6 anaconda
```

This may take a while to install, as it is replicating the default Anaconda packages in the new environment (alternatively, you can customise the packages installed, but this is not recommended).

Next, you must "activate" the new environment. While an environment is activated (in a given terminal window), we will be able to access the `python` executable and install any packages by simply calling `python` and `pip`. Note that activation is *slightly* different for Windows:

Windows:

```
$ activate python_3_gpss
```

Linux / macOS:

```
$ source activate python_3_gpss
```

You will now see that the terminal prompt is prefixed by `(python_3_gpss)`. While this is here, we are in the activated environment, and you can now install `GPy` as described above. Make sure to run the tests, as described, to confirm the installation.

Note that to deactivate the environment and return to the default Python environment, simply execute the command `deactivate` / `source deactivate` for Windows / Linux respectively.

Using the New Environment in Jupyter

By default, we cannot use the environment in Jupyter (and so cannot use it to run the labs). However, it is relatively simple to add the new environment as a "kernel" for Jupyter to use.

Simply activate the environment, and run the following command:

```
$ python -m ipykernel install --user --name python_3_gpss \
> --display-name "Python 3 (GPSS)"
$ deactivate
```

Remember that, for Linux or macOS, you must prepend `deactivate` with `source` .

We have deactivated, as we can now access the environment as a Jupyter kernel regardless of whether the environment is activated in the terminal -- this is particularly convenient.

If you run `jupyter notebook` now and open a notebook, you should be able to see the new kernel in the dropdown list when selecting a kernel. You should select your environment, the name of which will be shown in the top right of the notebook. **If you select the wrong kernel, or Jupyter chooses the wrong one, you can change it by selecting the correct one from the `Change kernel >` menu in the `Kernel` taskbar at the top.**

Deleting an Environment

If you need to delete a created environment, for example if you make a mistake and want to start afresh, you can simply use the `remove` command in `conda` :

```
$ conda remove --name python_3_gpss --all
```