COSC 2673 Machine Learning

Lab 01

Objective

- Learn about Anaconda, the data science/machine learning platform we will be using
- Learn some basic Python

Note at any time if you are familiar with a particular step, please feel free to skip forward to next one.

Introduction

We will be using Python as the programming language of the course. This lab is designed to introduce you to Python, iPython/Jupyter Notebooks and Anaconda, the Python environment and package manager we will using.

Note if you have installed Anaconda on your machines, then we suggest to use this rather than the core-teaching services, as the X-11 forwarding doesn't always work for the core teaching services.

If you have installed Anaconda on your own machines, please go to Part B of lab sheet. Otherwise go to Part A to study how to connect to the Core-teaching servers and use the version installed there.

Part A: Connecting to Anaconda and Python via the Coreteaching Servers

There are three CSIT servers used for teaching (the "coreteaching" servers):

- titan.csit.rmit.edu.au
- saturn.csit.rmit.edu.au
- jupiter.csit.rmit.edu.au

These are general purpose servers, running RedHat linux. The three servers share a common home directory space, so you can log in to any of the three servers, and should have access to the same files. (However, the three servers may have different loads -- if you are logged in to one and things seem to be running slowly, try to log in to another!)

A more extensive guide to the servers can be found here: https://docs.google.com/document/d/12CS_7OdEmpQZiwuxDTc9PHfHan4mGRwceT0t-kstKNc

COSC 2673 Machine Learning

Lab 01

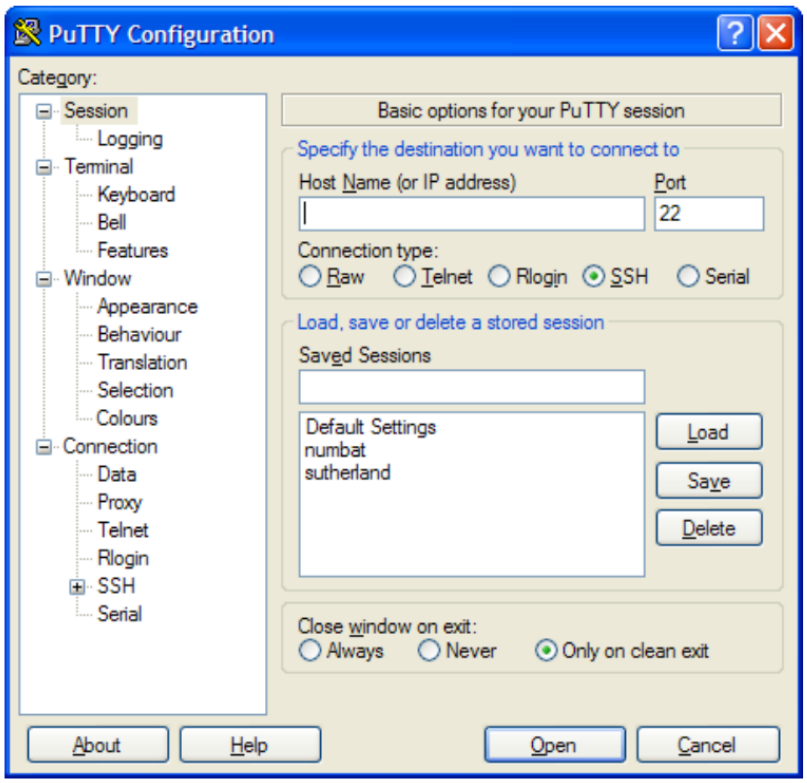
Your lab demonstrator should be able to help you if you run into difficulties, but if your account has not been set up correctly, you may have to contact ITS on 9925 8888, or from any landline at RMIT on 58888.

After your account has been setup, next step is to connect to these servers via ssh. This can be done in a variety of ways, depending on the operating system you are connecting from.

Connecting from a Windows PC, Using PuTTY and XMing

XMing is an open-source X11 display server for Microsoft Windows. In essence, it allows you to display unix/linux windows-based applications on a MS-Windows PC. To launch XMing, from the Windows Start menu scroll down to the folder item "XMing", and select the "XMing" item within. This will launch a process that you should leave running in the background.

PuTTY is an open-source terminal emulator that you can use to securely connect to a unix/linux machine. To launch PuTTY, from the Windows Start menu scroll down to the folder item "PuTTY", and select the "PuTTY" item within. You will see a new window similar to the following.



First, select the "SSH" option in the "Category" block on the left (it's the second item from the bottom). Find the checkbox labelled "Enable X11 forwarding", and select it. This enables the core teaching servers to display graphical interfaces over ssh.

Then, select the "Session" option in the "Category" block. The screen will look similar to the screenshot above. You can now enter the name of a coreteaching server in the "Host Name (or IP address)" item in the middle block. Type in one of the coreteaching server names, such as titan.csit.rmit.edu.au

It may be handy to save this session information so that you can create subsequent connections more quickly; in the "Saved Sessions" box, type a name such as "titan", and then click the "Save" button on the right. (You will see that "titan" gets added to the list of saved sessions in the middle of the screen; when you open PuTTY in the future, you can simply double click this name to restore your settings to connect to titan.)

Now click the "Open" button. The first time that you connect to a particular server, you will be a message such as the one to the right. Click yes.

A new window will then open, with a live connection to the coreteaching server. The first thing it will ask you is: "login as:". Here, you need to enter your userid (s1234567) and press enter, then enter your password.

Congratulations, you are now logged in to a coreteaching server, and can start entering commands at the command prompt.

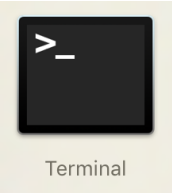
Further information on connecting to the coreteaching servers, and getting started with Unix/Linux, is available in the CSIT Unix Survival Guide. **If you haven't yet read this document, now would be a good time to do so!**

Connecting from a Mac, Using the OSX Terminal

To run X-windows applications in OSX, you will first need to install XQuartz (an open source X server). Once you have installed XQuartz, it will launch automatically when an X application is invoked through the terminal (see next step).

The OSX Terminal utility is a terminal emulator that provides command-line (i.e. text-based) access to the operating system. This is convenient for many things, including running many

programs and tools, and to connect to other machines. In OSX Sierra (and most previous versions), Terminal is found in the "Utilities" folder, within the "Applications" folder.



To connect to a coreteaching server, launch a Terminal window, and then open a secure shell (SSH) connection by entering the following commands at the terminal prompt:

```
$ ssh -X s1234567@server.csit.rmit.edu.au
```

where s1234567 should be replaced with a string consisting of the letter "s" followed by your student number, and server should be one of titan, saturn, or jupiter, depending on which coreteaching server you want to connect to.

After a few moments, you may be asked whether you want to add a key (this should only happen the first time that you connect from a particular machine -- type "yes"), after which you will be prompted for your password. You should see information similar to the following:

```
$ ssh s1234567@titan.csit.rmit.edu.au
s1234567@titan.csit.rmit.edu.au's password: <enter password>
Last login: Thu Jul 16 14:29:45 2017 from csitprdap01.int.its.rmit.edu.au

              *** PLEASE READ CAREFULLY ***
      ***** This service is for authorised users only *****

  Individuals using this computer system without authority, or in
  excess of their authority,

  ... [snip] ...

*******************************************************************************
*  WARNING: It is a criminal offence to:                                      *
*       i.  Obtain access to data without authority                           *
*       ii. Damage, delete, alter or insert data without authority            *
*******************************************************************************

  ... [snip] ...

$
```

Congratulations, you are now logged in to a coreteaching server, and can start entering commands at the command prompt, $.

Further information on connecting to the coreteaching servers, and getting started with Unix/Linux, is available in the CSIT Unix Survival Guide. **If you haven't yet read this document, now would be a good time to do so!**

Connecting from a Linux maching, using Terminal

Similar to the MAC instructions, you'll need to ensure X forwarding is active on your system, then type in the following command:

```
$ ssh -X s1234567@server.csit.rmit.edu.au
```

where s1234567 should be replaced with a string consisting of the letter "s" followed by your student number, and server should be one of titan, saturn, or jupiter, depending on which coreteaching server you want to connect to.

Setting Up iPython on the Coreteaching Servers

In this course, we will be making extensive use of the iPython/Jupyter interactive environments. On the coreteaching servers, iPython is part of the anaconda distribution, which is installed in the following location:

```
/opt/anaconda/2/v4.2.0/bin/ipython
```

Therefore, to launch iPython, type:

```
$ /opt/anaconda/2/v4.2.0/bin/ipython
```

You may wish to save yourself some typing in the future by adding the anaconda path to your local profile. To do so, you will need to use your favourite text editor to edit your .bash_profile configuration file. Suppose that your favourite text editor is nano, type:

```
$ nano .bash_profile
```

Near the top of your .bash_profile file, you will see a line that includes the word PATH. It may look similar to the following (but will almost certainly not be identical!):

```
PATH=$PATH:$HOME/bin
```

You need to extend this line, by adding a colon (:) followed by the anaconda directory, to the end. Do not include any spaces. For example, if the above was initially in your .bash_profile, you would edit it to read:

```
PATH=$PATH:$HOME/bin:/opt/anaconda/2/v4.2.0/bin
```

Save your .bash_profile, exit the text editor, and source your settings by entering:

```
$ source .bash_profile
```

Now, you should be able to launch iPython by simply typing

```
$ ipython
```

at the command prompt.

Optional: Create directory for lab exercises

First, let's create a directory for machine learning, and this tutorial.

After you log in to a coreteaching server, you should start in your home directory. If you already changed to another directory, simply type

```
$ cd
```

to return to your home directory.

Now, type

```
$ mkdir pds
```

to create a new directory for machine learning. Change into this directory using the "cd" (change directory) command, and then create a new directory under it, called "lab01", and change to this directory:

```
$ cd pds
$ mkdir tute01
$ cd tute01
```

You should now be in your new "tute01" directory. From here, you can launch iPython. You should see some information, followed by an iPython prompt:

```
$ ipython
Python 2.7.12 |Anaconda 4.2.0 (64-bit)| (default, Jul  2 2016, 17:42:40)
Type "copyright", "credits" or "license" for more information.

IPython 5.1.0 -- An enhanced Interactive Python.
?         -> Introduction and overview of IPython's features.
```

```
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.
```

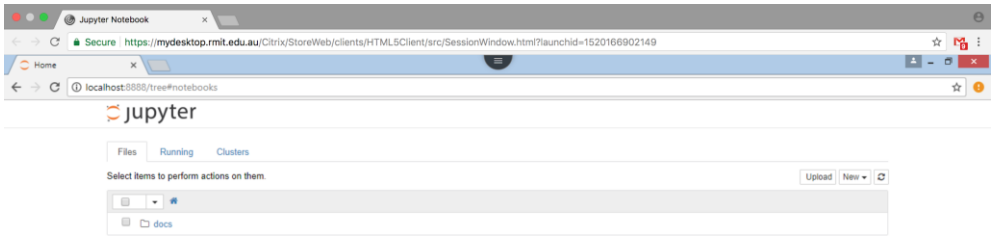
We are now ready to learn and try out some Python commands.

Skip Part B and go to Part C to learn about some basic Python commands.

Part B: Graphical Anaconda - Accessing Jupyter Notebook

Once you have installed Anaconda on your own machines, start Anaconda and select Jupyter Notebook, and interactive python environment. You can access it via:

Then, you will see the following (or a similar) screenshot:



Then,

- Select New -> Python 2 (or Python 3 if you prefer but remember we the syntax we use in class will be focused on Python 2)
- The new created '*.ipynd' is typically created at the following location (but can differ on your machine):
    - C:\Users\sXXXXXXX
    - where sXXXXXXX should be replaced with a string consisting of the letter "s" followed by your student number.

More materials for Jupyter Notebook:

- A tutorial:
  - Jupyter Notebook Tutorial: The Definitive Guide (https://www.datacamp.com/community/tutorials/tutorial-jupyter-notebook)

More materials for Anaconda:

- Install Install Anaconda on your PC:
  - https://www.anaconda.com/download
  - (please select Python 2.7 version)

Optional: Create directory for lab exercises

First, let's create a directory for machine learning, and this tutorial.

After you log in to Lab computer, you should start in your home directory. You can create a new folder named "pds" for Practical Data Science.

Then create a new directory under it, called "tute01", and save your *.ipynb files (from Jupyter Notebook) here for this tutorial.

We are now ready to learn and try out some Python commands.

Part C: A First iPython Session

For data exploration, it is often convenient to work in an interactive environment. iPython offers such an environment for the Python language.

Now enter the following commands to print out "Hello Machine Learning, here I come!":

```
In [1]: print("Hello Machine Learning, here I come!")
```

This should output the string to standard output.

Python is an interpreted language, and you can use it as such like a calculator:

```
In [2]: 1 + 2
```

What does it display?

COSC 2673 Machine Learning

Lab 01

Python can store values into variables, which are assigned a type according to what is assigned to it. Type in the following:

```
In [3]: a = “Hello World”
```

This stores the string “Hello World” in the variable a. We can print that out as follows:

```
In [4]: print(a)
```

In addition to strings, Python has other types, such as integers:

```
In [5]: b = 3 * 4
```

This will store the integer ‘12’ into b.

Python has two major builtin data structures (there are other useful ones we will use for machine learning, but for now it is important to understand the built-in ones), one is a list and another is a dictionary. We will focus on list for now (please read up about dictionaries). A list In Python is essentially a sequence of elements, that we can reference using indices (thing of this as similar to a Java array, but can dynamically increase in size). First we create a list:

```
In [6]: c = [1,2,3]
```

Creates a list with 3 elements of 1, 2, 3. To access an element, we use an index. Python list index starts from 0:

```
In [7]: c[0]
```

What does this display?

Finally, to exit your ipython session, simply type

```
In [8]: exit
```

For the rest of the lab and to learn Python, please go through the following or online resources described in the Machine Learning Canvas shell.

You can learn more about iPython from the official documentation. There are also many tutorials available online.

Introductory guide that those new to programming or need a refresher:
https://www.programiz.com/python-programming

If you familiar with programming and scripting languages in general, consider looking at this site, which goes through the major parts of Python:
https://www.stavros.io/tutorials/python/