Lab Course: Distributed Data Analytics

0. Overview

Mofassir ul Islam Arif

Information Systems and Machine Learning Lab (ISMLL)
Institute for Computer Science
University of Hildesheim, Germany

April 8, 2019

Outline

0. Organizational Stuff
1. Lecture Overview
2. Introduction to Python
3. Numpy, Scipy, Pandas and matplotlib
4. Reading Material and Softwares

Outline

Exam and Credit Points (1/2)

- The course gives 6 ECTS
- requires 180h student effort, the duration of the course is 14 weeks.
  1. 4h/week (in the lab)
  2. 9h/week (own time for solving exercise sheets)
  3. $(4 + 9)$ h/w * 14 w $= 180$h

- There will be a weekly exercise sheet.
- You will get approximately 6 to 7 days in-between the date of release and the date of submission.
- The grading of this course will be based on solutions submitted in each individual lab.
  - There will be no written exam at the end of term

Exam and Credit Points (2/2)

- The course can be used in
  - Data Analytics MSc
  - IMIT and AINF MSc. / Informatik / Gebiet KI & ML
  - Wirtschaftsinformatik MSc / Business Intelligence
- Register yourself at LSF (POS module) and learnweb.
- `https://www.uni-hildesheim.de/learnweb2019/course/search.php?search=3116`
- Enrollment key is 3116
- Withdrawl from the lab is ONLY possible until the 5th Exercise submission.

Exercises

- There will be a weekly exercise sheet with 3 questions uploaded **every Friday** to learnweb (3116).
- Solutions to the exercises can be submitted until **next Friday 23:59 Berlin Time**
- Solutions will be discussed in next Lab, Students will present their work
- Labs Group 1 **every Monday 14:00–18:00**, C-147: 2nd Sem +
- Labs Group 2 **every Thursday 10:00–14:00**, C-147: 1st Sem
- Each lab exercise will carry equal weight-age towards the final mark.

Excercise Submission Format

Each Excercise will consists of the following questions

- Q1: Implement a given model in python [10-12 Marks]
  - Need to provide complete **working** code
- Q2: Show learning properties of model/algorithm [5-8 Marks]
  - Graphs showing learning curve
  - explanation of the graphs/tables
- Q3: Solve problem with state-of-the-art library [5 Marks]
  - Graph comparing state-of-the-art and your code
  - Comparison of execution time (etc)
- **Submission must include**:
  - Code Files (1 for each task), zipped
  - PDF file with analysis and graphs

Exercise Checking

- Each student will submit an individual solution. (no group submissions)
- All submissions should be made through the learnweb (course code 3116).
- No late submission, missing a lab will result in 0 points.
- Points will be awarded based on your submitted report and code.
- To obtain maximum mark, Your work needs to stand out as compared to your peers.
  - Working code **doesn't** mean full points. That is the **minimum** requirement
- A question answer session (Lab viva) will be conducted for a random sample of students.
- Write your own code/solution. Do not copy it.

Plagiarism

Plagiarism is

- to steal and pass off (the ideas or words of another) as one's own
- to use (another's production) without crediting the source
- to commit IP theft
- to present as new and original an idea or product derived from an existing source

0% tolerance for Plagiarism

Consequence includes

- ZERO to all parties involved
- Referral of the case to the exam branch
- Exam Branch exmatriculates the parties involved
- A Fail grade in the degree, not just the lab

Meeting

My Office hours

Tuesdays
12:00 - 14:00
C206
SPL
or by Appointment
email: mofassir@ismll.de

Outline

Syllabus

Thu. 09.04.	(1)	Introduction and Distributed Computing with MPI I
Thu. 16.04.	(2)	Distributed Computing with MPI II
Thu. 23.04.	(3)	Distributed Computing with MPI III
Thu. 30.04.	(4)	TensorFlow I
Thu. 07.05.	(5)	TensorFlow II
Thu. 14.05.	(6)	TensorFlow III
Thu. 21.05.	(7)	TensorFlow III
Thu. 28.05.	(8)	Apache Spark I
Thu. 04.06.	(9)	Apache Spark II
Thu. 11.06.	(10)	Apache Spark III
Thu. 18.06.	(11)	Distributed Machine Learning Algorithm I
Thu. 25.06.	(12)	Distributed Machine Learning Algorithm II
Thu. 02.07.	(13)	Distributed Machine Learning Algorithm III

Outline

Getting Started

Installing python: two possible ways

- Directly install python from python.org
  - Ubuntu: $apt-get install python
  - $pip install <packages> (pip is a python package installation utility)
  - $ python (launch python shell)
  - $ python script.py (run python script)
- Install Anaconda platform (most of the packages are pre-installed)
  - Follow the instructions:
    `https://docs.anaconda.com/anaconda/install/linux/`
  - $ jupyter notebook (a interactive web based python shell)
  - $ ipython (launch python shell)

Installing on Windows

Installing python on windows

- Click Here: `https://docs.anaconda.com/anaconda/install/windows/`
- Follow the instructions
- Should be straight forward from there

Python Basics (1/6)

- Python is an interpreted language like PHP or Perl
- Python is interactive and allows programming to interact with the interpreter
- Python is Object-Oriented language i.e. supports concepts of encapsulation
- Python is easy to learn (also known as beginner's language)
- Python is portable, scalable

Python Basics (2/6)

- The zen of python (type *import this*)
- **White Space formating:**
  - Python uses indentation to delimit a block of code i.e.

```
1 for i in range(1,10):
2   for j in range(11,20):
3     print(i+j)
4   print(i)
5 print('End of For Loop')
6 varA = 1 + 3
```

  - Generally blackslash is used to indicate a statement continues onto the nextline

Python Basics (3/6)

- **Modules**
  - All the features/modules that you may require are not loaded by default
  - To load a module: *import <package> as alias*
  - Or explicitly load: *from <package> import <subpackage> as alias*

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from collections import Counter
4
```

- **Counter**
  - A Counter is a dict subclass and is used for counting hashable objects

```
1 from collections import Counter
2 numbers = [0, 1, 3, 1, 0, 1]
3 c = Counter(numbers)   #Counter({0: 2, 1: 3, 3: 1})
4
```

Python Basics (4/6)

- **Lists and Tuples:**
  - Lists in python are mutable (can be changed)
  - Tuples are closer to lists but are immutable object (readonly)

```
positive = list(range(10))
list1 = [1, 2, 2, 1, 5, 2, 3]
list1.append(3)
prime = (1,3,5,7,11,13) #cannot add elements

```

- **Dictionaries and Sets:**
  - Dictionaries are key-value pair, allows quick access.
  - Sets represents a collection of *distinct* elements
  - Sets are itself mutable but can only hold immutable objects

```
d1 = dict()
grades = {''Joe'': 80, ''Tim'' : 90 }
g1 = grades[''Joe'']
grades[''Alice'']   # return keyError
 s = set(list1) # {1, 2, 3, 5}

```

Python Basics (5/5)

- **Functions:**
  - Syntax:

```
1 def function_name(parameters):
2   ''' function Doc String '''
3   function suite
4   return [expression] # not mandatory
5
```

- **Control Statements**
  - *if-elif-else* , *while* and *for* provde control statements

```
1 if condition1:
2   statements
3 elif condition2:
4   statements
5 else
6   statements
7
```

Outline

Numpy, Scipy, Pandas and matplotlib

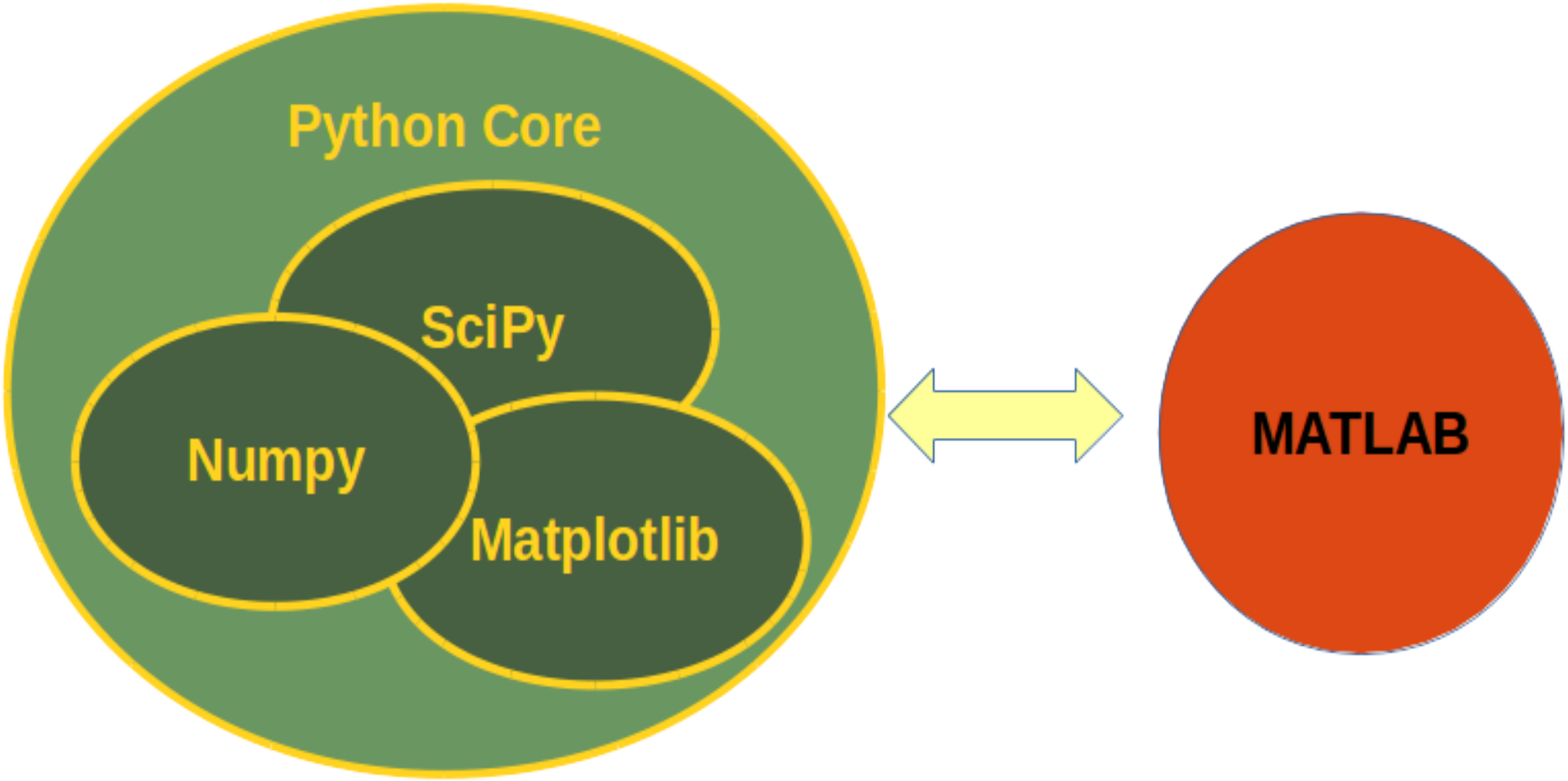


Figure: reference:
`http://www.python-course.eu/numerical_programming.php`

Numpy (1/4)

Numpy is an extension of python, adding support for large, multi-dimensional arrays object and associated routines for fast operations on them.

```
1  import numpy as np
2  a = np.arange(15).reshape(3,5)
3  b = np.array([[1.0, 2, 3.0], [2.0, 3, 2]])
4  c = np.arange(3)**2 # ** is a power operator
5  d = np.random.random((2,3))
6  x = np.linspace( 0, 2*np.pi, 100 )
7  f = np.sin(x)
8  f[1:5]  #array([ 0.06342392,  0.12659245,  0.18925124])
9  f[-3:-1] # equal to f[97:99]
10
```

- also see: array, zeros, empty, arange, linspace, rand, randn
- argmax, argmin, argsort, average, median, sort, outer, prod

Numpy (2/4)

Reshaping array

```
a = np.floor(10*np.random.random((3,4)))
a.shape # (3,4)
a.ravel() # flatten the array
a.shape = (6, 2)
a.reshape(3,-1) # with -1, the other dimension is
    automatically calculated
np.vstack(a,b)  #  stack columns, or np.hstack(a,b) for rows
np.hsplit(a,2) # reverse of stacking
b = arange(12)**2
j = array( [ [ 3, 4], [ 9, 7 ] ] )  # a bidimensional array
    of indices
a[j] # same shape

```

- also see: array, zeros, empty, arange, linspace, rand, randn
- argmax, argmin, argsort, average, median, sort, outer, prod

Numpy (3/4)

Numpy and Linear Algebra

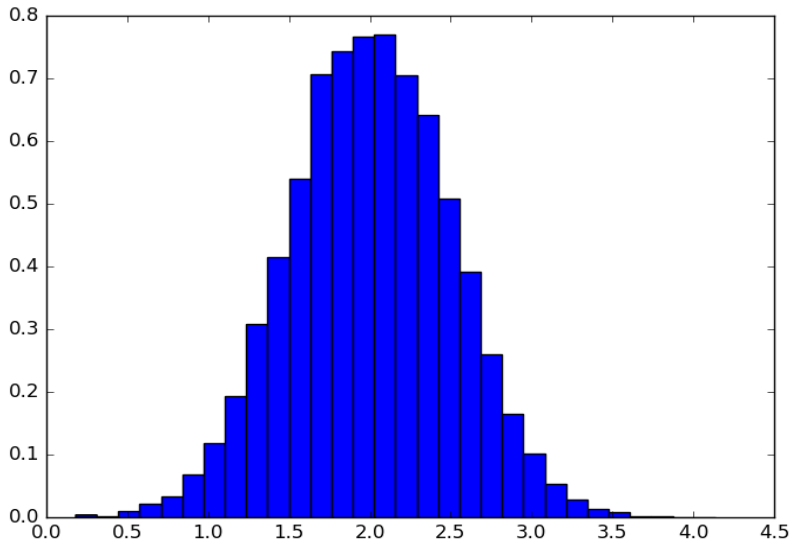
```
1 import numpy as np
2 import numpy.linalg as linalg
3 a = np.array([[1.0, 2.0], [3.0, 4.0]])
4 y = np.array([[5.], [7.]])
5 a.transpose() # a.trace(), np.inv(a)
6 linalg.solve(a,y) #  help(linalg.solve)  to know more about a
      method
7 a[:,1] # create a slice of original array a. Slice is another
      view of same object
8
9
```

- inv, svd, norm, eig, eye, qr, lstsq, tensorsolve, tensorinv

Numpy (4/4)

Histogram with matplotlib

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 mu, sigma = 2, 0.5
4 v = np.random.normal(mu,sigma,10000)
5 plt.hist(v, bins=50, normed=1) # matplotlib version (plot)
6 plt.show()
7
```



Pandas

```
1  import numpy as np
2  import pandas as pd
3  import matplotlib.pyplot as plt
4  from scipy import stats
5  # must specify that blank space " " is NaN
6  data = pd.read_csv("/home/user/parasite_data.csv", na_values
       =[" "])
7  data.head() # shows top 5 rows and tail() shows botton 5 rows
8  data.fillna(0.0).describe()  # data.describe()
9  # with and without ignoring NaN values
10 print ("Mean:", data["Virulence"].mean())
11 print ("Mean w/ filled NaN:", data.fillna(0.0)["Virulence"].
       mean())
12 plt.hist(data.fillna(0.0)["Virulence"], bins=5, normed=1)
13
```

1) download data `https://github.com/rhiever/ipython-notebook-workshop/blob/master/parasite_data.csv`

Outline

Some Books

- Kevin P. Murphy (2012):
  Machine Learning, A Probabilistic Approach, MIT Press.
- Joel Grus (2015):
  Data Science from Scratch First Principles with Python, O'Reilly
- Wes McKinney (2012):
  Python for Data Analysis Data Wrangling with Pandas, NumPy, and IPython, O'Reilly
- Willi Richert, Luis Pedro Coelho (2013):
  Building Machine Learning Systems with Python, PACKT

Some Useful Tutorials

- Python 3
  `http://www.python-course.eu/python3_course.php`
- Numerical and Scientific Programming with Python
  `http://www.python-course.eu/numerical_programming.php`
  `https://docs.scipy.org/doc/numpy-dev/user/quickstart.html`
- Basic to Advance Python
  https://www.tutorialspoint.com/python/index.htm
- Pandas
  `http://www.gregreda.com/2013/10/26/intro-to-pandas-data-structures/`
- Matplotlib: Plotting
  1) `http://www.scipy-lectures.org/intro/matplotlib/matplotlib.html`

Some Machine Learning Software

- Python (v3.5, v2.7; `https://www.python.org/`).
- Anaconda (4.2.0 (Python v3.7, v2.7); `https://www.anaconda.com/distribution/`).
  with Anaconda you will get most of the libraries and software pre-installed
- TensorFlow (`https://www.tensorflow.org`)
- scikit-learn (v0.17; `http://scikit-learn.org/stable/index.html`)

Public data sets:

- UCI Machine Learning Repository (`http://archive.ics.uci.edu/ml/`)
- UCI Knowledge Discovery in Databases Archive (`http://kdd.ics.uci.edu/`)