Department of Computer Engineering
University of Peradeniya

Data Mining and Machine Learning
Lab 01

June 15, 2017

1 Introduction

NumPy, short for Numerical Python, is the fundamental package required for high-performance scientific computing and data analysis. Most of the frameworks and libraries such as Scikit-Learn, Tensorflow which require numerical calculation use NumPy.

1.1 Creation

```
import numpy as np # import numpy module as np

matrix = np.array([np.arange(3), [i for i in range(3,6)], [6,7,8]])
matrix
Out[14]:
array([[0, 1, 2],
       [3, 4, 5],
       [6,7,8]])
```

1.2 Initialization

```
np.zeros([4,5,6]  # create an array of zeros
np.ones([3,4], dtype=float) # create an array of ones
np.arange(1,10,1).reshape(3,3) # create an array evenly spaced values
np.linspace(0,2,9) # create an array of zeros
np.full([3,3], 4) # create a constant array
np.eye(3) # create an identity matrix
```

1.3 Copying, Sorting, Slicing

```
np.copy(matrix)
matrix.copy() # deep copy
matrix.view() # shallow copy
matrix.sort() # return a sorted copy of an array reference to same
#location.
```

```
matrix.sort(axis=1)
matrix[0:,:1]
matrix[:2, 1:]
matrix[:2, :]
```

1.3.1 Try Out

```
matrix[1,0]
matrix[0] = 42
matrix[1:3]
matrix[]
matrix[1:]
matrix[1:100]
matrix[:]
matrix[1:,:2]
matrix[:2, 1:]
matrix.ravel()
matrix[:,1].copy()
matrix[1].tolist()
matrix.reshape(-1)
```

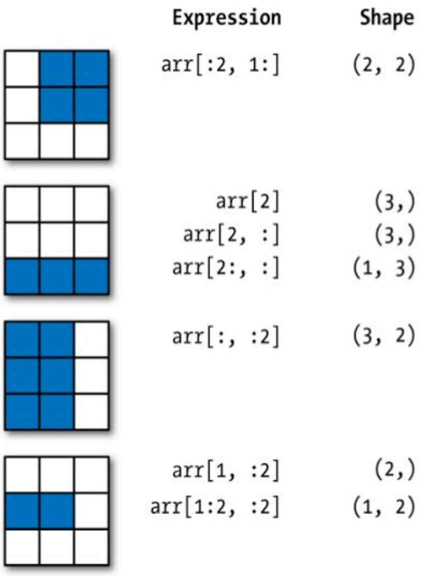

Figure 1: Two dimensional array slicing

Now see Fig. 1 for an illustration.

1.4 Apply elements wise operations and functions

1.4.1 Try Out

```
np.sqrt(matrix)
np.exp(matrix)
np.min(np.maximum(np.random.randn(6), np.random.randn(6)))
np.mean(matrix)
np.mean(matrix, axis=0)
np.sum(matrix)
np.invert(matrix)
np.random.randint(0, 1)
```

Hope by now, you have the basic understanding about how to deal with NumPy. Try to solve this problem. Let's assume you are to implement basic version of random walk. Walk would start at any point(e.g. 0 or 10) step of 0 or 1. Current position should be deducted by 1 for 0 value occurrence. Try to implement single random walk with 500 steps.

2 Python Classes

2.1 Class definition syntax

```
class RandomWalk:
    <statement-1>
    .
    .
    .
    <statement-N>
```

2.2 Class Objects

```
class RandomWalk:
    def __init__(self, position):
        self.position = position

    def walk(self):
        return walked_path

random_walker = RandomWalk(200)
```

3 Lab Exercise

3.1 Find the value of s_N

s_N can be defined as $\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i-\bar{x})^2}$ where number of rows in provided matrix is denoted by N, $\bar{x}$ is the mean of a given column and x_i is the ith element of a given column. Use $labExercise01$ dataset for this.

3.2 Bonus Exercise

Develop a small algorithm which can be used to detect the local minima of given vector. To test your algorithm use *bonusExercise* dataset and to visualize your result you may use below code. Visualization is not expected.

```
import matplotlib.pyplot as plt
## TODO calculate local minima
local_minina = detect_local_minima(args...)
plt.scatter(np.array(local_minina)[:, 0], np.array(local_minina)[:, 1],
color=’red’, label="Local Minima")
plt.show()
```

4 Submission

Submit a single .py file. Rename it as [12|13]xxxlab01.py where xxx is your registration number.

5 Important

Make sure that now you have the basic understand what we did during the lab. This lab is really important for successive labs as well. If you do not understand any concepts, make sure you get some help from instructors.

6 Deadline

The deadline: June 29, 23:59:59 GMT+5:30.