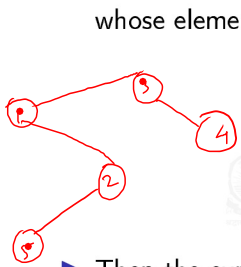
Graph Properties

Objectives

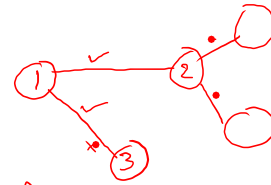
- Network Properties Using Adjacency Matrix

Node Degree

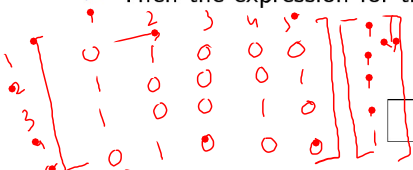
- Let A be an adjacency matrix of an undirected graph and $\mathbf{1}$ be the column vector whose elements are all one.



$$A_{ij} = \begin{pmatrix} a_{11} & \dots & \dots & a_{1n} \\ a_{21} & \dots & \dots & a_{2n} \\ a_{31} & \dots & \dots & a_{3n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & \dots & \dots & a_{nn} \end{pmatrix}, \quad \mathbf{1} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$$

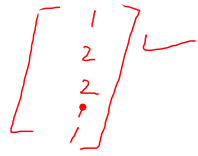


- Then the expression for the vector K whose elements are degree k_i of the vertices:



$$K = A\mathbf{1}$$

We are adding columns of A



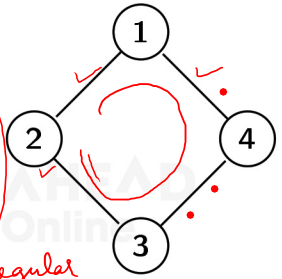
EXAMPLES

2-REGULAR GRAPH

K- regular → Each node has degree K

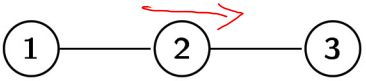
$$A\mathbf{1} = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 2 \\ 2 \\ 2 \end{pmatrix} \begin{pmatrix} 3 \\ 3 \\ 3 \\ 3 \end{pmatrix}$$

3-regular



PATH GRAPH

$$A\mathbf{1} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}$$

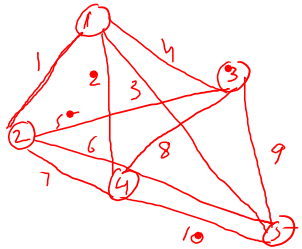


Each node • 4 degree

4 regular graph

5 vertices

How many edges ?



Handshaking theorem

10 edges.

Sum of degree of all vertices = 2 • E

N • K = 2 • E

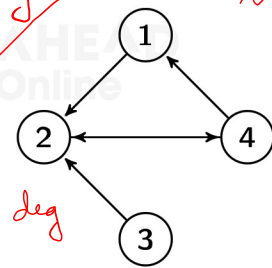
E = N • K / 2 = 10

DIRECTED GRAPH

- Let A be an adjacency matrix of an directed graph and $\mathbf{1}$ is a column vector of 1's.
- The vector $K = A\mathbf{1}$ represents out-degree of k_i vertices and the vector $K = A^T\mathbf{1}$ represents in-degree of k_i vertices.

$$A\mathbf{1} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 2 \end{pmatrix}$$

$$A^T\mathbf{1} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 3 \\ 0 \\ 1 \end{pmatrix}$$

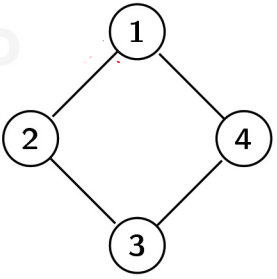


Number of Edges

- Let A be an adjacency matrix of an undirected graph and $\mathbf{1}$ be the column vector whose elements are all one.
- Then, the number of m edges in the network is $= \frac{1}{2}\mathbf{1}^T A\mathbf{1}$

Example

$$\frac{1}{2}\begin{pmatrix}1 & 1 & 1 & 1\end{pmatrix}\begin{pmatrix}0 & 1 & 1 & 0\\ 1 & 0 & 1 & 0\\ 1 & 1 & 0 & 1\\ 0 & 0 & 1 & 0\end{pmatrix}\begin{pmatrix}1\\1\\1\\1\end{pmatrix} = \frac{1}{2}\,8 = 4$$



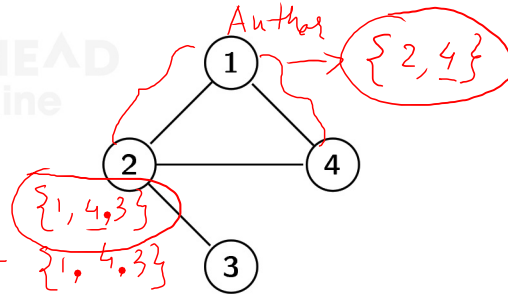
Number of m edges for a directed graph is $A\mathbf{1}$ or $A^T\mathbf{1}$

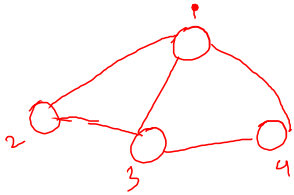
Number of Common Neighbors

- Let A be an adjacency matrix of an undirected graph
- Then, the matrix N whose elements N_{ij} is equal to number of common neighbors of vertices i and j is A^2

Example

$$\begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 2 & 1 & 1 & 1 \\ 1 & 3 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 2 \end{pmatrix}$$



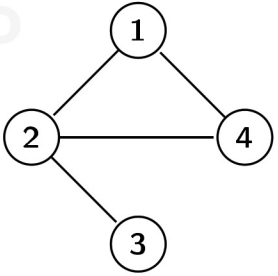

1
2
3
4

TRACE OF THE MATRIX

- The trace of a square matrix A, denoted $tr(A)$, is defined to be the sum of elements on the main diagonal
- Notice that, $tr(A)$ is $2m$, where m is the number of edges

EXAMPLE

$$\begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix} = \begin{bmatrix} 2 & 1 & 1 & 1 \\ 1 & 3 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 2 \end{bmatrix}$$

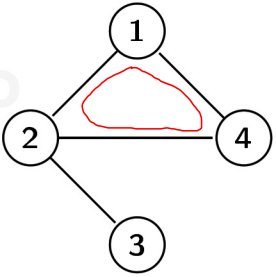


Number of Triangles

- The number of triangles in the network is $\frac{1}{6}tr(A^3)$

Example

$$\underbrace{\begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} 2 & 1 & 1 & 1 \\ 1 & 3 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 2 \end{pmatrix}}_{A^2} = \begin{bmatrix} 2 & 4 & 1 & 3 \\ 4 & 2 & 3 & 4 \\ 1 & 3 & 0 & 1 \\ 3 & 4 & 1 & 2 \end{bmatrix}$$



No. of Triangles in the network $= \frac{1}{6}\,6 = 1$

Summary

Adjacency Matrix

- Node Degree
- Edges
- Common Neighbors
- Triangles

Web graphs

OBJECTIVES

Basics of World Wide Web

Webgraph

World Wide Web

WWW

- World Wide Web, also known as a Web
- A collection of websites or web pages stored in web servers
- Web pages are formatted in HTML and connected by links called "hypertext" or hyperlinks and accessed by HTTP protocol

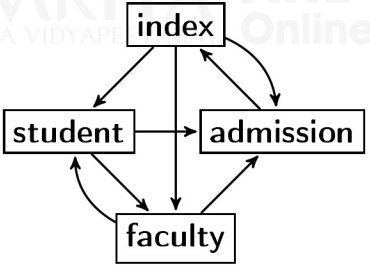
Defintion

- The World Wide Web (WWW, or simply Web) is an information space in which the items of interest, referred to as resources, are identified by global identifiers called Uniform Resource Identifiers (URI).

WEBGRAPH

BASICS

- Webgraph is a directed graph
- Nodes in the graph are web pages of WWW
- Edges are hyperlinks between pages



Web Charactristics

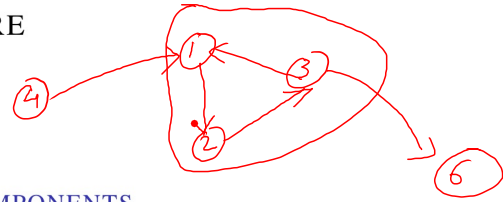
Basics

- W3C https://www.w3.org/
- The World Wide Web Consortium (W3C) is the main international standards organization for the World Wide Web
- Attempted to answer questions like: How many web pages are there? and How fast is the Web growing?

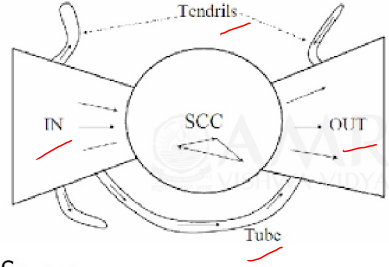
Some characteristics

- How it look like as a graph ?
- Size/Number of web sites, pages
- Distribution of web sites by country, language, ..
- Popular web sites (to a community)
- Popular web sites by in-links

Web Structure



Graph Structure



Source:

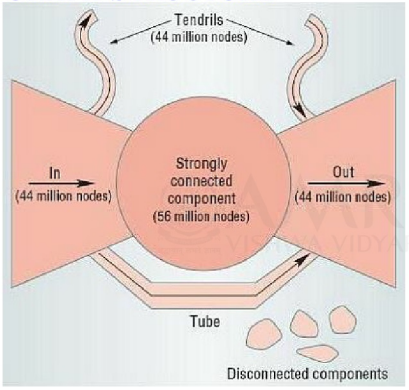
`https://nlp.stanford.edu/IR-book/html/htmledition/the-web-graph-1.html`

Components

- SCC — Strongly Connected Component
- IN
- OUT
- Tendrils
- Tubes
- Disconnected

Web Characterization

Graph Structure

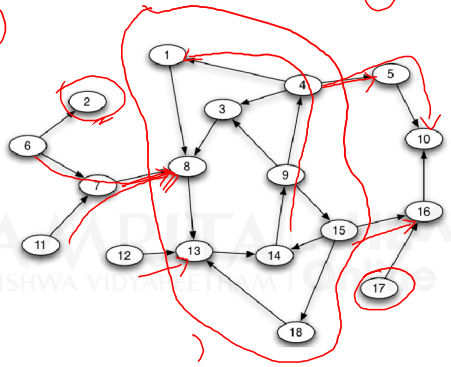


Source: `https://users.dimi.uniud.it/~massimo.franceschet/networks/nexus/pagerank.html`

Components

- A strogly connected component **(SCC)** contains pages between which users can surf easily.
- Another large cluster, labelled **'in'**, contains pages that link to the core but cannot be reached from SCC.
- A separate **'out'** cluster consists of pages that can be reached from the SCC but do not link to it, such as corporate websites containing only internal links.
- Other groups of pages, called **'tendrils'** and **'tubes'**, connect to either the in or out clusters, or both, but not to the core, whereas some pages are completely unconnected

EXAMPLE



[1]

SCC: {1, 3, 4, 8, 9, 13, 14, 15, 18}

IN: {6, 7, 11, 12}

OUT: {5, 10, 16}

Tendrils: {2, 17}

Tubes: { }

Disconn: { }

Summary

Web Graph

- World Wide Web
- Structure of Web

Objectives

- Concepts of Citation Networks
- Cocitation Matrix and Networks

CITATION NETWORKS

Bibliometrics
Scientometrics

- Citation networks are the network of citations between academic papers.
- A citation graph or network is a directed graph describes citations within a collection of documents.
- Nodes in the grpah represents **documents** and a directed edge represents a **citation** from one document to another.

P1 → P2

P1 cites P2

CONSTRUCTING CITATION GRAPHS

Papers refer to one or more other previous papers, usually in a bibliography at the end of the paper, and one can construct a network in which the vertices are papers and there is a directed edge from paper A to paper B if A cites B in its bibliography.

Note that a citation network is acyclic, while the Web is not

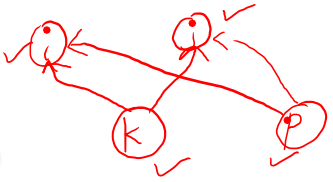
P2 cites P1

Cocitation Graphs

- Adjacency matrix of a citation graph is same as that of a directed graph.

$$A_{ij} = \begin{cases} 1, \text{ if there is an edge from } j \text{ to } i \\ 0, \text{ otherwise} \end{cases}$$

- We take the opposite convention for ease of computation.



Cocitation Concepts

- The cocitation of two vertices i and j in a directed network is the number of vertices that have outgoing edges pointing to both i and j.
- The cocitation of two papers is the number of other papers that cite both.

$$A_{ik}A_{jk} = \begin{cases} 1, \text{if } i \text{ and } j \text{ are both cited by } k \\ 0, \text{ otherwise} \end{cases}$$

COCITATION MATRIX

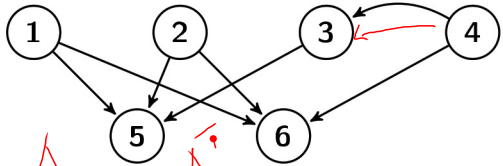
- Summing over all k, the cocitation C_{ij} of i and j is

$$C_{ij} = \sum_{k=1}^{n} A_{ik} A_{jk} = \sum_{k=1}^{n} A_{ik} A_{kj}^{T}$$

- The *cocitation* matrix C is a $n \times n$ matrix with elements C_{ij} , which is thus given by

$$C = AA^{T}$$

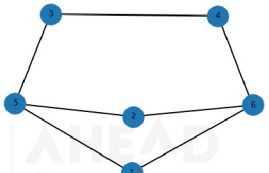
EXAMPLE



$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 2 \\ 0 & 0 & 0 & 0 & 2 & 2 \end{pmatrix}$$

Cocitation Network

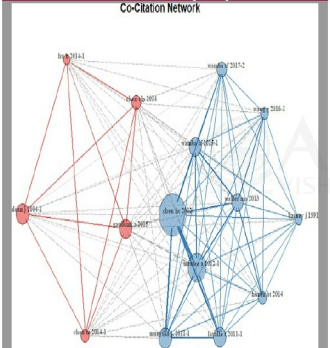
$$C = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 2 \\ 0 & 0 & 0 & 0 & 2 & 2 \end{pmatrix}$$



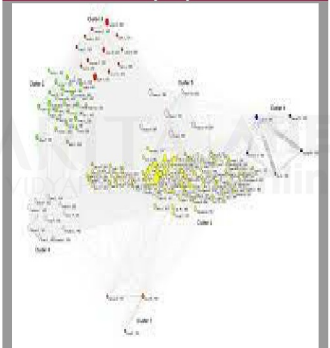
- Note that C is a symmetric matrix, $CT = (AA^T)^T = AA^T = C$
- Also, a strong cocitation between papers is often a good indicator of papers that deal with related topics

RESEARCH IN COCITATION NETWORKS

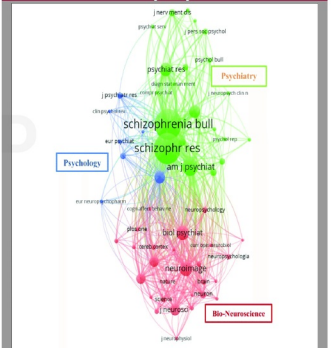
COCITATION [2]



MAPPING [3]



CITATION [1]



Summary

Cocitation Network

- Concepts of Cocitation
- Cocitation Matrix

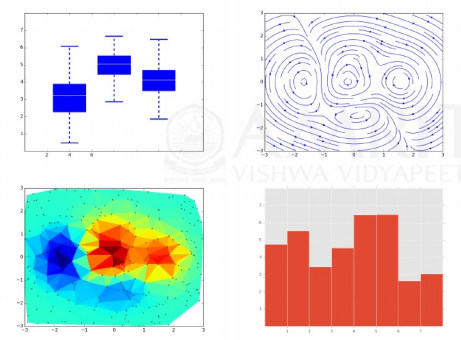
References

- A bibliometric analysis of six decades of research
  www.researchgate.net/figure/Network-resulting-from-the-co-citation-analysis-by-journal-Co-citation-fig3_323285144.
- Citation network
  www.sciencedirect.com/topics/computer-science/citation-network.
- Mapping recent information behavior research: an analysis of co-authorship and co-citation networks
  www.researchgate.net/figure/Co-citation-network_fig4_272998649.
- Newman, M.
  Networks.
  Oxford university press, 2018.

Matplotlib

OBJECTIVES

- Introduction to Matplotlib



Matplotlib

- Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.
- A low level graph plotting library
- Matplotlib utilities lies under the pyplot submodule, and are usually imported under the *plt* alias

```
import matplotlib.pyplot as plt
```

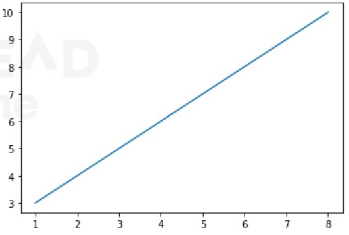
Features

- Interactive figures
- Export to several file formats
- Can integrate with Jupyter

A Simple Example

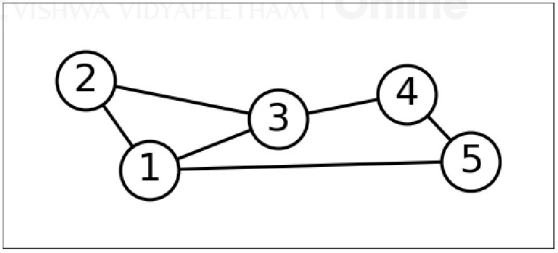
- The plot() function is used to draw points (markers) in a diagram.
- By default, the plot() function draws a line from point to point.

```
import matplotlib.pyplot as plt
import numpy as np
xpoints = np.array([1, 8])
ypoints = np.array([3, 10])
plt.plot(xpoints, ypoints)
plt.show()
```



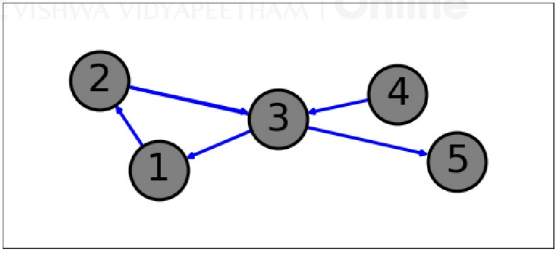
DRAWING IN NETWORKX

```
import networkx as nx
import matplotlib.pyplot as plt
G1 = nx.DiGraph( [(1,2),(1,3),(1,5)] )
G1.add_edges_from([(2, 3), (3, 4),(4, 5)])
nx.draw(G1, with_labels = True)
plt.show()
```



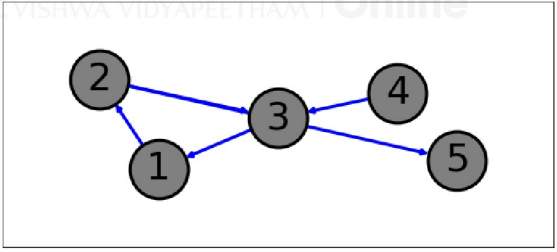
DRAWING IN NETWORKX ..

```
import networkx as nx
import matplotlib.pyplot as plt
G1 = nx.DiGraph( [(1,2),(2,3),(3, 1)] )
G1.add_edges_from([(4, 3),(3, 5)])
nx.draw(G1, with_labels = True)
plt.show()
```



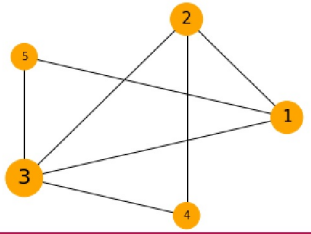
SAVING PLOTS

```
import networkx as nx
import matplotlib.pyplot as plt
.......................
.......................
plt.savefig("graph1.png")
plt.savefig("graph1.pdf")
```



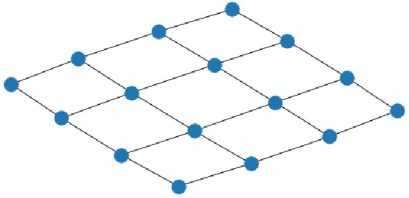
MANY OPTIONS

```
from matplotlib.pyplot import figure, text
import matplotlib.pyplot as plt
G=nx.Graph()
e=[(1,2),(1,5),(2,3),(4,2),(4,3),(3,5),(1,3)]
G.add_edges_from(e)
pos = nx.circular_layout(G)
figure(figsize=(4,3))
d = dict(G.degree)
nx.draw(G, pos=pos,node_color='orange',
        with_labels=False,
        node_size=[d[k]*300 for k in d])
for node, (x, y) in pos.items():
    text(x, y, node, fontsize=d[node]*5, ha='center', va='center
        ')
plt.axis("off")
plt.savefig("plot4.png")
```



MANY MORE

```
import matplotlib.pyplot as plt
import networkx as nx
G = nx.grid_2d_graph(4, 4)  # 5x5 grid
# print the adjacency list
for line in nx.generate_adjlist(G):
    print(line)
# write edgelist to grid.edgelist
nx.write_edgelist(G, path="grid.edgelist",
    delimiter=":")
# read edgelist from grid.edgelist
H = nx.read_edgelist(path="grid.edgelist",
    delimiter=":")
pos = nx.spring_layout(H, seed=200)
nx.draw(H, pos)
```



Summary

Matplotlib

- Basic plotting
- Drawing Networks
- Some features

References


Hagberg, A., Swart, P., and S Chult, D.
Exploring network structure, dynamics, and function using networkx.
Tech. rep., Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.


Hunter, J. D.
Matplotlib: A 2d graphics environment.
Computing in Science & Engineering 9, 3 (2007), 90–95.

OBJECTIVES

Python for Data Science

Anaconda

About

- Anaconda Distribution is a Python/R data science distribution
- Includes a package and environment manager
- Tools to easily collect data from various sources

Conda

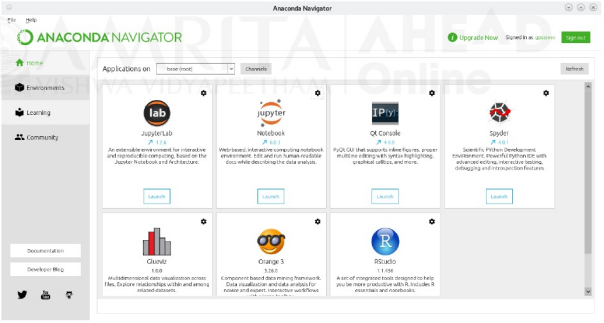
conda is a tool for managing and deploying applications, environments and packages

```
conda info
conda list
conda search PACKAGENAME
conda install PACKAGENAME
```

Anaconda Navigator

About

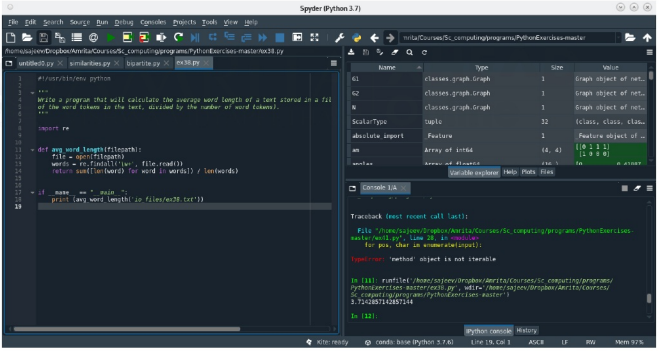
- Anaconda Navigator is a desktop graphical user interface (GUI)
- Allows you to launch applications and easily manage conda packages, environments, and channels without using command-line commands



Spyder

About

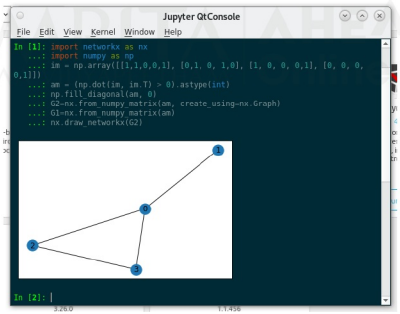
- Spyder, the Scientific Python Development Environment
- It includes editing, interactive testing, debugging, and introspection features.



Qt Console

About

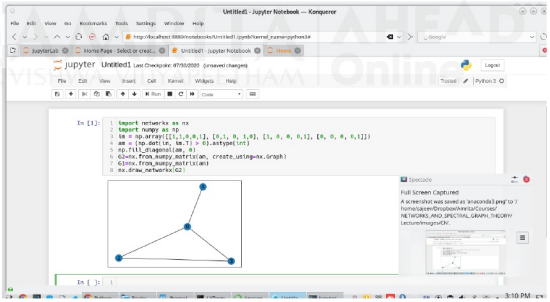
- The Qt console is a very lightweight application like a terminal.
- Powerful with inline figures and syntax highlighting
- The Qt console can use any Jupyter kernel.



Jupyter Notebook

About

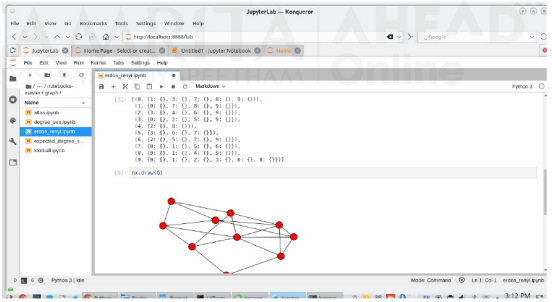
- The notebook extends the console-based approach to interactive computing
- A web-based for developing, documenting, and executing code, as well as communicating the results



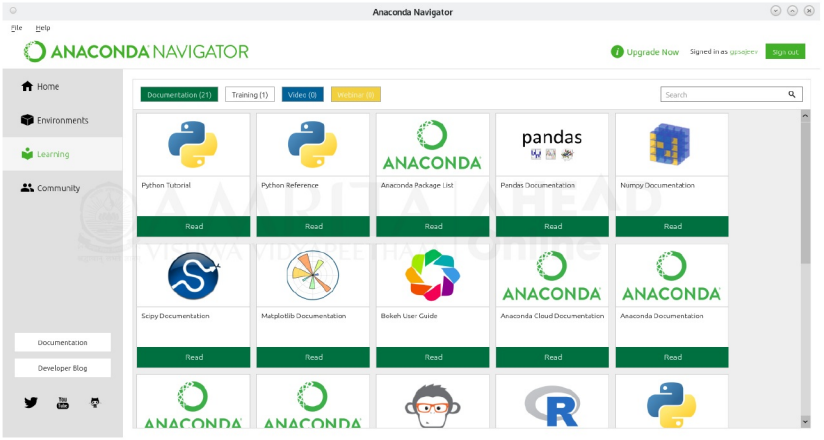
Jupyter Lab

About

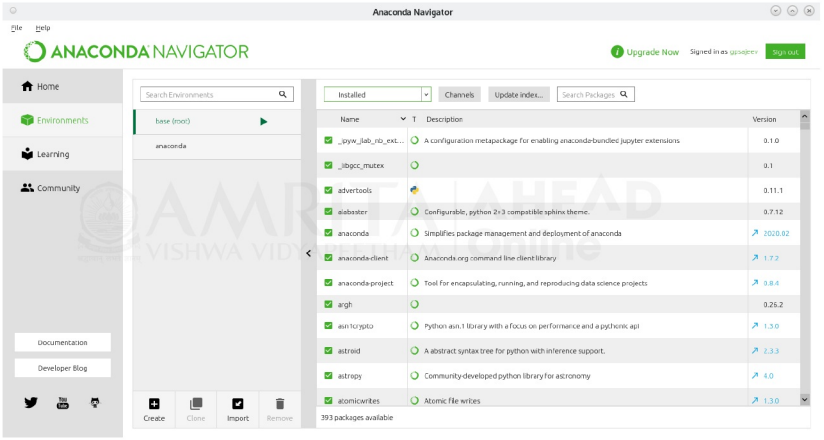
- An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.
- Jupyter Lab extensions are available



Learning in Anaconda Navigator



ANACONDA ENVIRONMENT



Summary

Anaconda

- Conda
- Navigator
- GUIs
- Help System

References

Anaconda software distribution.
Anaconda Documentation, Anaconda Inc., *https://docs.anaconda.com/* (2020).

OBJECTIVES

- Using *Jupyter Notebook* and *JupyterLab*

Jupyter Notebook

- The notebook combines live code, equations, narrative text, visualizations
- Leverage big data tools, such as Apache Spark, from Python, R, and Scala. Explore that same data with pandas, scikit-learn, ggplot2, and TensorFlow.
- Jupyter supports over 40 programming languages, including Python, R, Julia, and Scala.
- Code can produce rich, interactive output: HTML, images, videos, LaTeX, and custom MIME types

JUPYTER NOTEBOOK

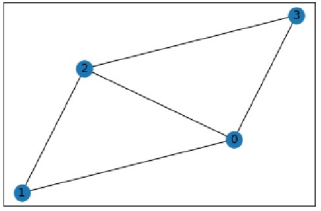
- A Jupyter notebook is made up of a number of cells.
- Each cell can contain Python/R code.
- Code cells, Markdown cells and Raw cells

```
[ ]: import networkx as nx
     G = nx.Graph()
```

Working with Notebook

- We can execute a particular cell by pressing *Shift-Enter*, or clicking on "play" button

[15]:
```
G1 = nx.Graph( [(0,1),(2,3),(3,0)] )
#G1.add_edge(0, 3)
G1.add_edges_from([(0, 2), (2, 1)])
nx.draw_networkx(G1)
```



Working with Notebook ..

- If the cell is a Markdown cell, the markdown text will get rendered beneath the cell.
- For contents in the raw cells, there will be no rendering

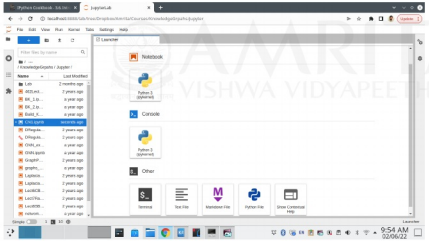
[25]:
```
# Accessing edges and neighbors
In addition to the views ‘Graph.edges()‘, and ‘Graph.adj()‘,
access to edges and neighbors is possible using subscript notation.
```

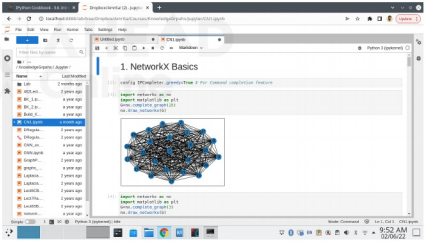
Accessing edges and neighbors

In addition to the views `Graph.edges()`, and `Graph.adj()`, access to edges and neighbors is possible using subscript notation.

JupyterLab

- JupyterLab is the next generation of the Jupyter Notebook
- JupyterLab uses the exact same Notebook server and file format as the classic Jupyter Notebook





SUMMARY

JUPYTER NOTEBOOK & LAB

References

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S., and Willing, C.
Jupyter notebooks – a publishing format for reproducible computational workflows.