Mining and Learning with Graph

1. How do we mine networks?

(1) Empirically:

Study network data to find organizational principles.

How do we measure and quantify networks?

(2) Mathematical models:

Graph theory and statisical models.

Models allow us to understand behaviors and distinguish surprising from expected phenomena.

(3) Algorithms for analyzing graph

Hard computational challenges.

2. What do we study in networks?

(1) Structure and evolution:

What is the structure of a network?

Why and how did it come tu have such structure?

(2) Processes and dgnamics

3. Structure of Networks?

(1)Network:

A network is a collection of objects where some pairs of objects are connected by links.

(2)Components of a Network:

Object: nodes, vertices N

Interactions(相互作用): links, edges E

System: network, graph G(N, E)

(a pair of a node set and edge set)

(3)Networks&Graph:

Network:often refers to real systems

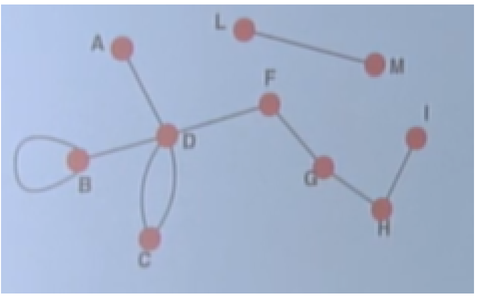
Graph: is a mathematical repersentation of a network

(4) Representations

*Graph Types

a.Undirected(无向)

Links:undirected(symmetrical, reciprocal)



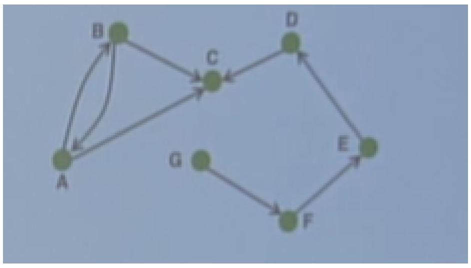
Example:

collaborations

friendship on facebook

b.Directed(有向)

Links：directed（arcs）



Examples:

phone calls

following on Twitter

Node Degrees:

Undirected:

Node degree(Ki):

the number of edges adjacent to node i.

Avg. degree:

$$\bar{k} = \langle k \rangle = \frac{1}{N}\sum_{i=1}^{N} k_i = \frac{2E}{N}$$

Directed:

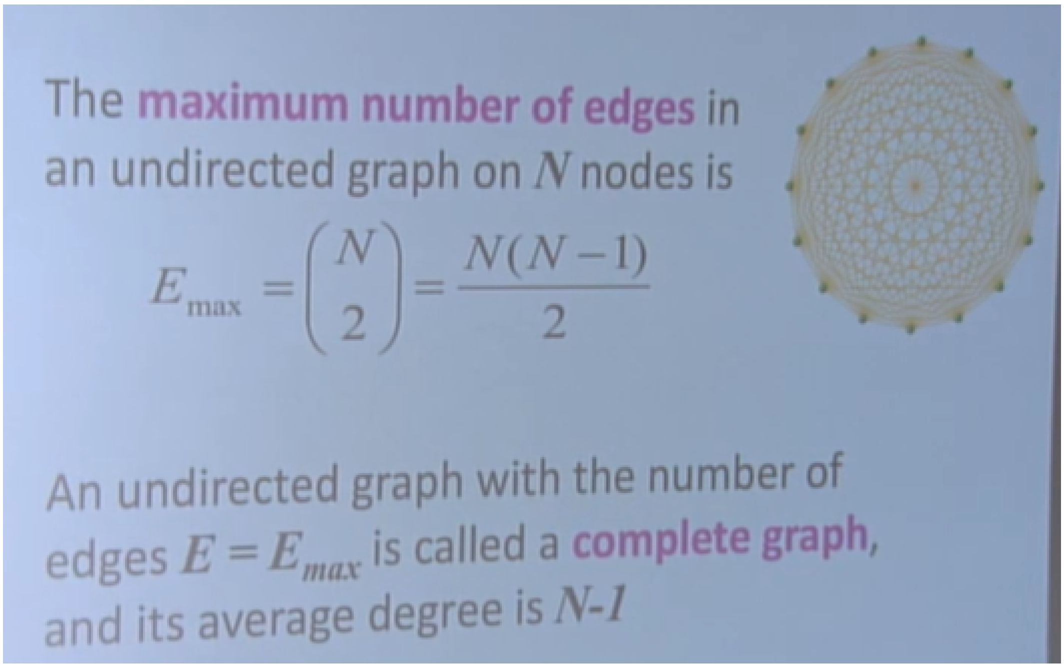
In-degree&Out-degree

The totoal degree of a node is the sum of in-and out-degrees.

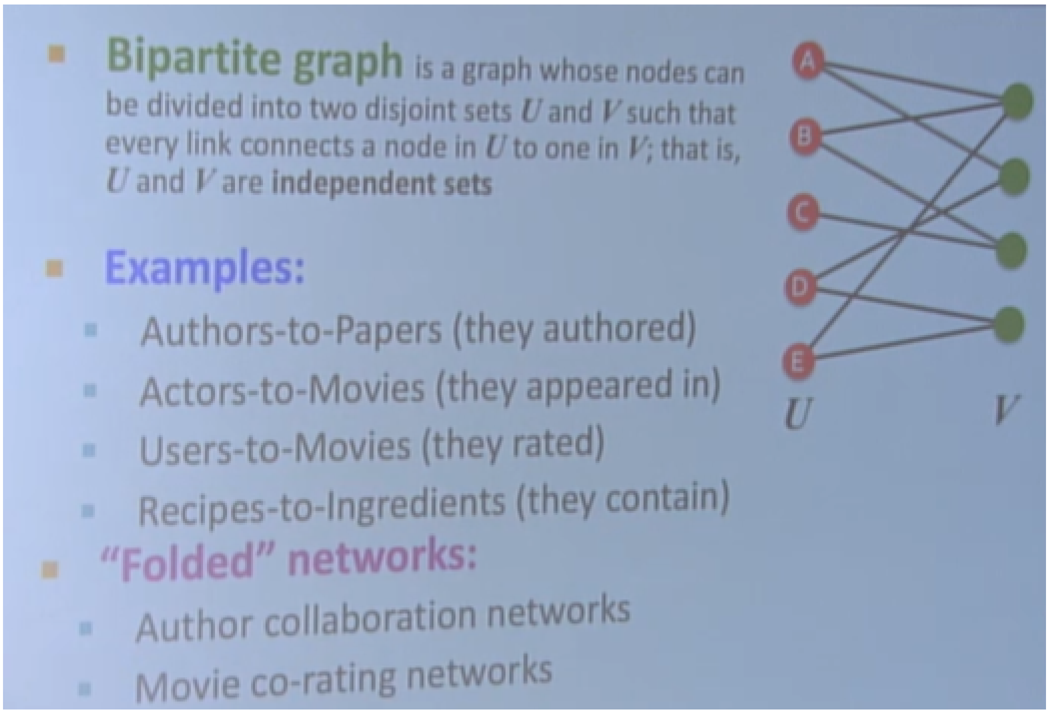
K_in=K_out

$$\bar{k} = \frac{E}{N}$$

c. Complete Graph

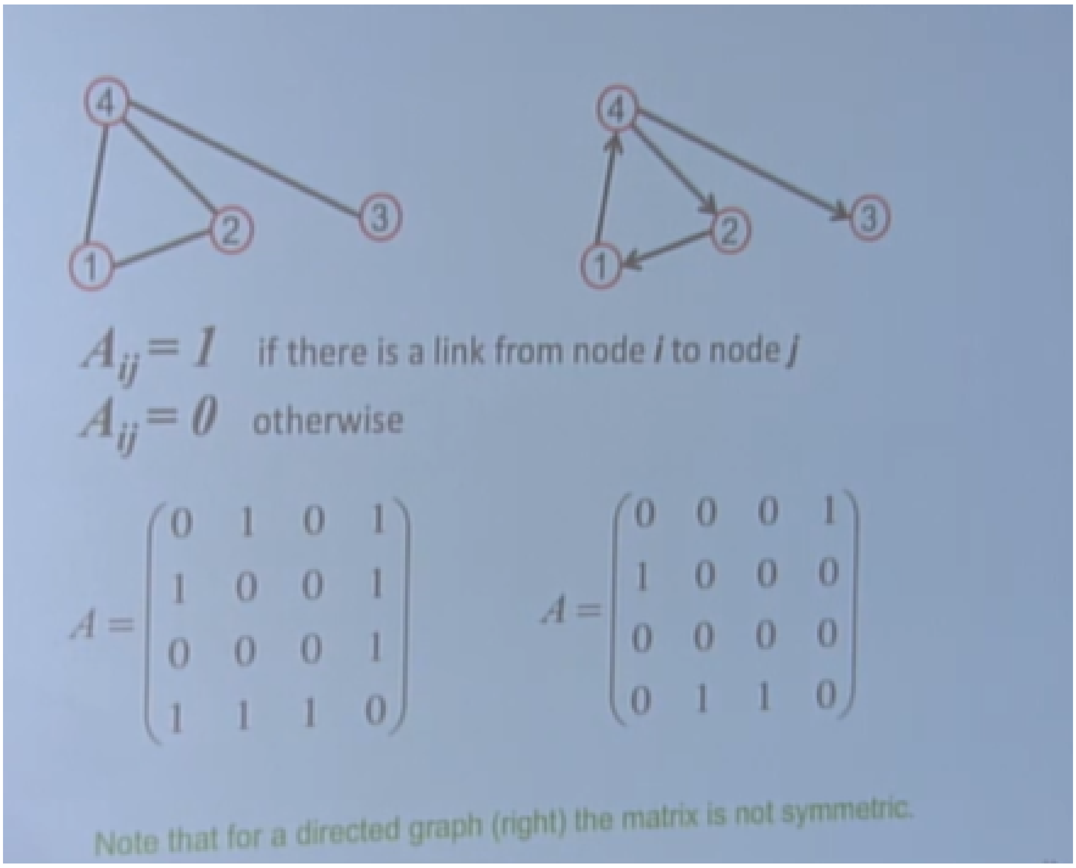


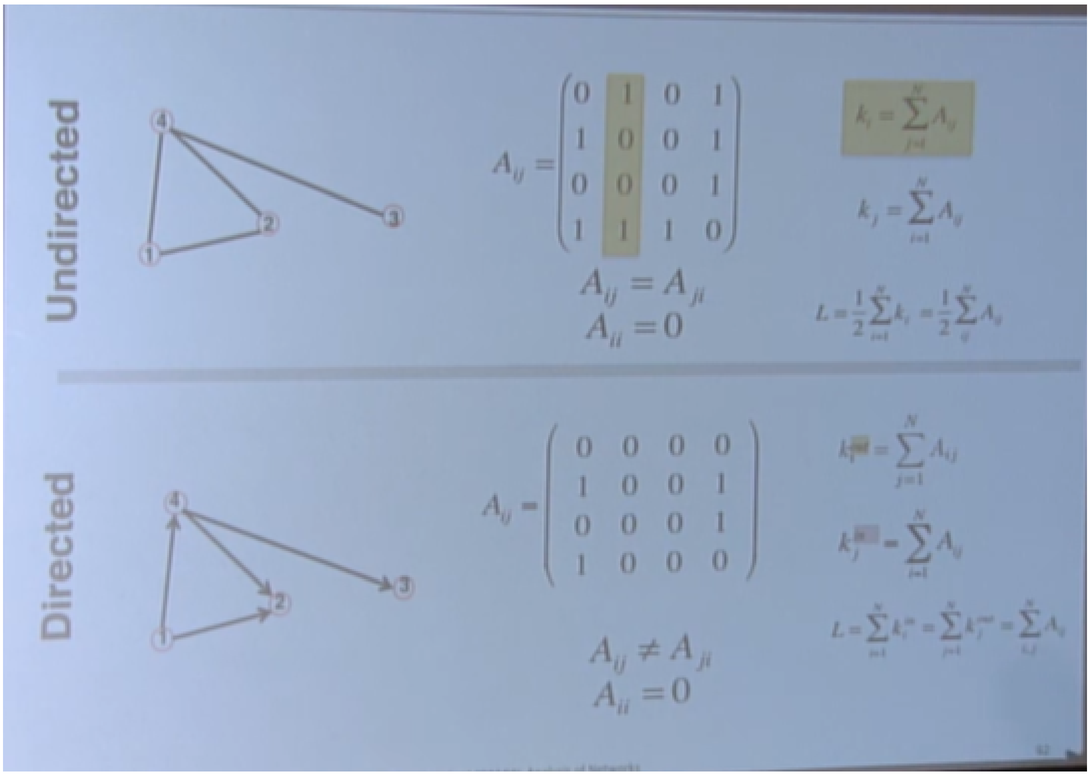
d. Bipatite Graph



* Representing Graphs:

a.Adjacency Matrix(邻接矩阵)

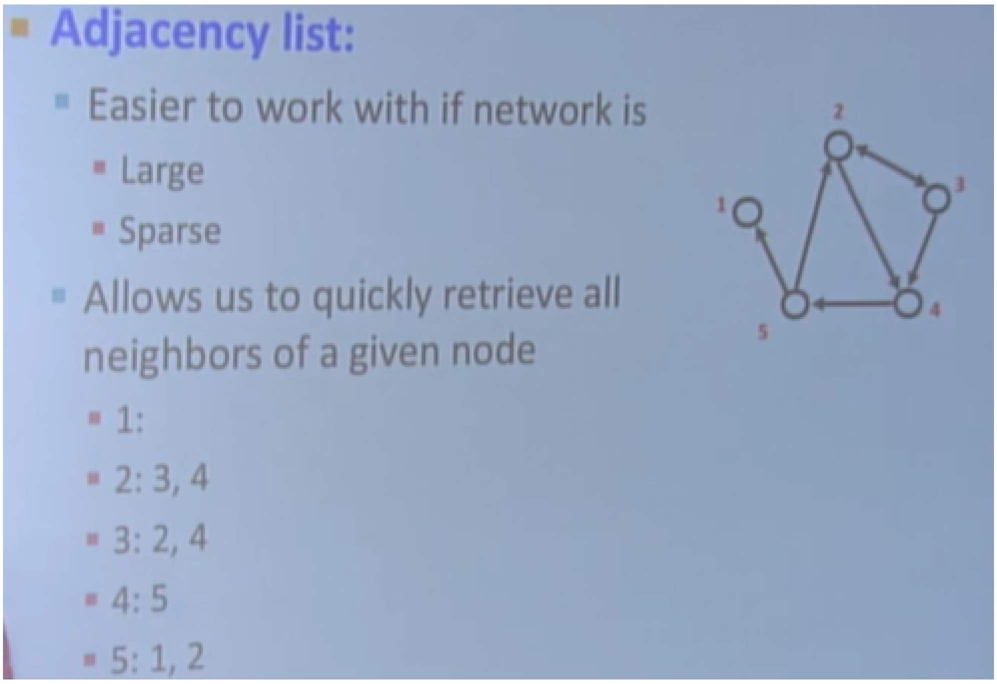




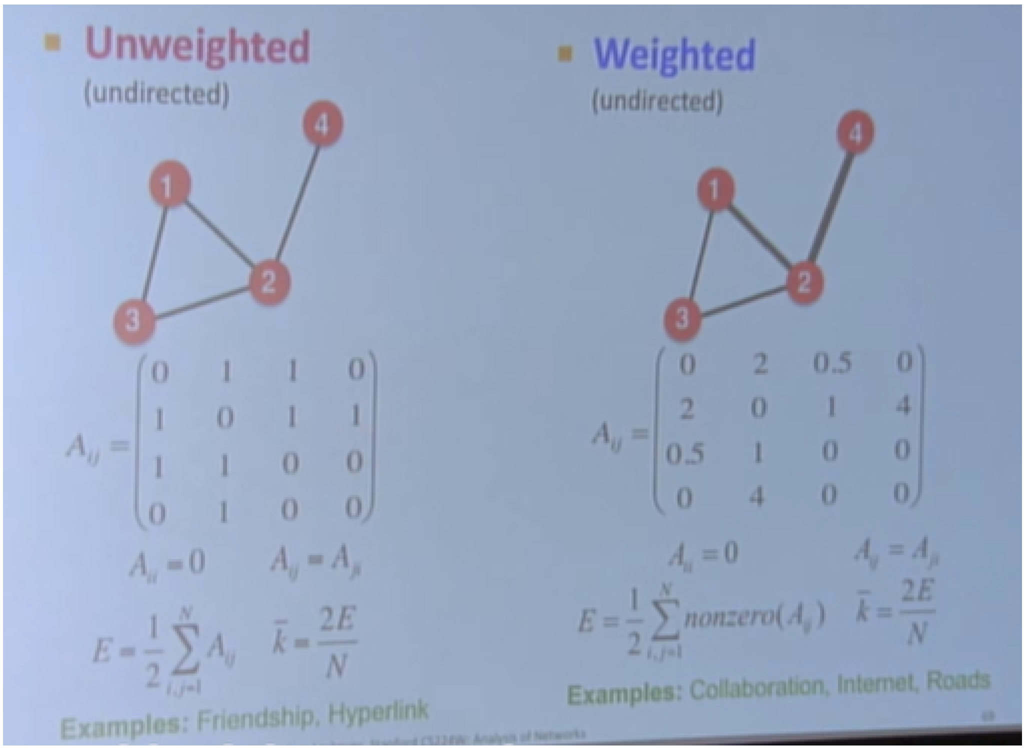
Adjacency Matrices are Sparse(稀疏)

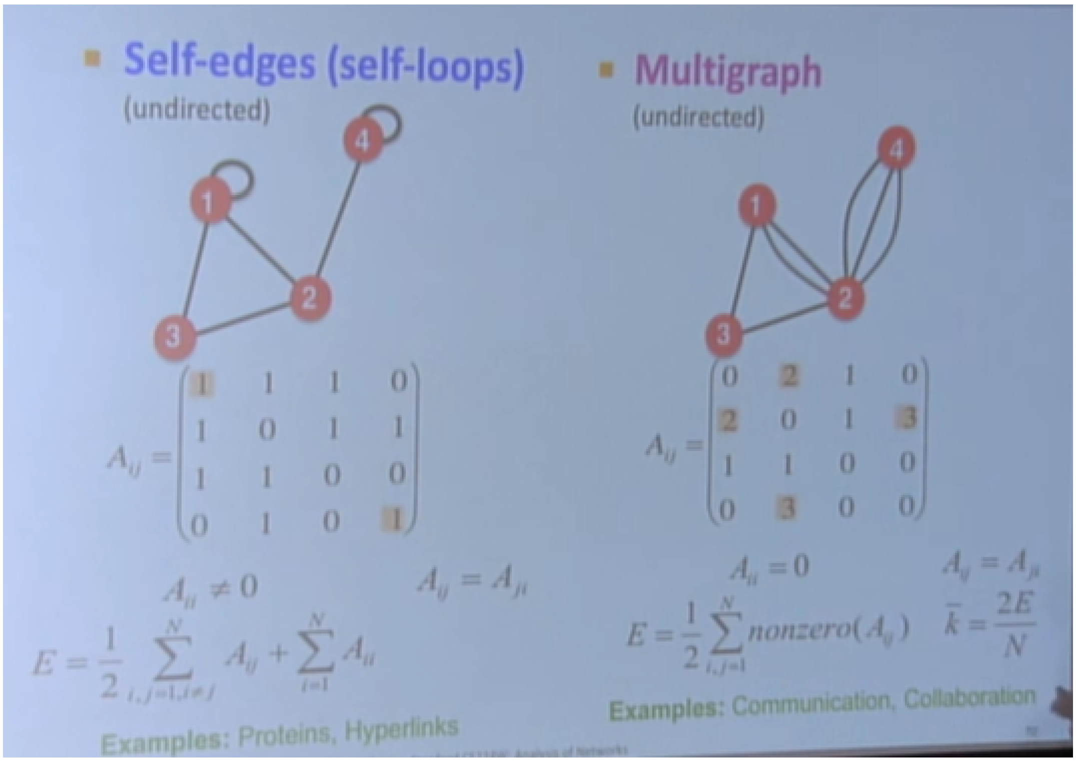
b.Adjacency list

represent graph as a set of edges.



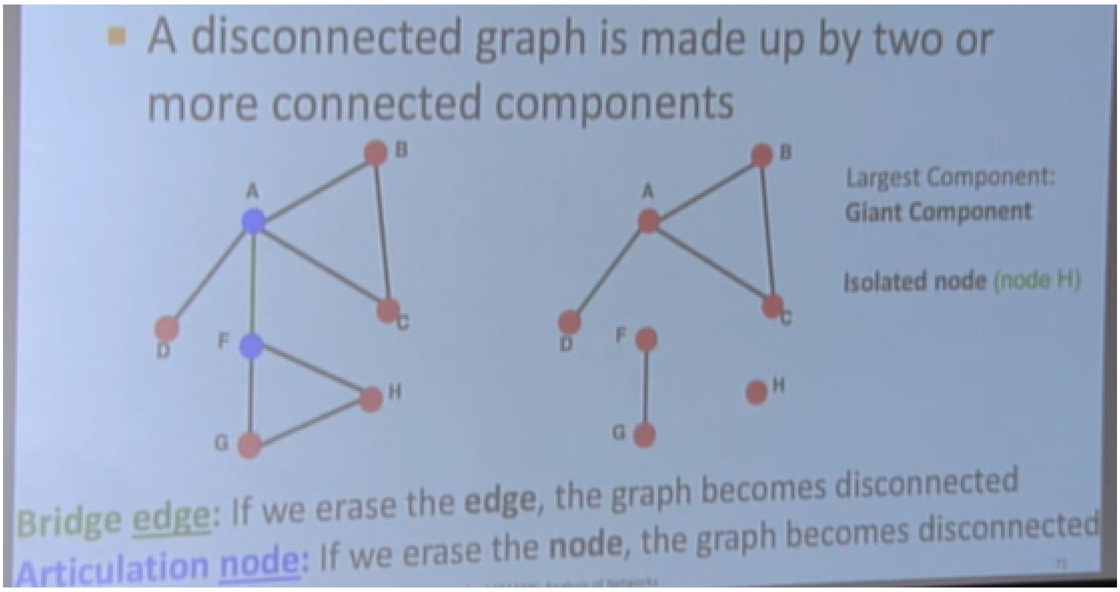
c.Weighted&Unweighted





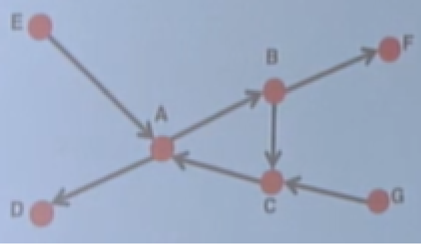
(5)Connectivity of Undirected Graphs

Connected(undircetde) graph:any two vertices can be joined by a path.



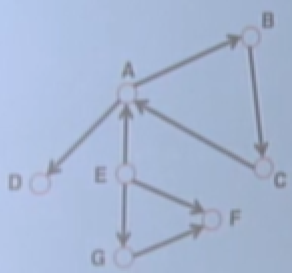
(6) Connectivity of Directed Graphs

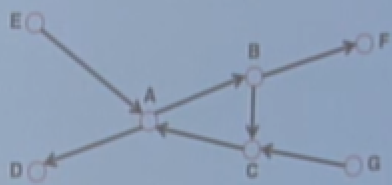
- **Strongly connected directed graph**
  - has a path from each node to every other node and vice versa (e.g., A-B path and B-A path)
- **Weakly connected directed graph**
  - is connected if we disregard the edge directions



Graph on the left is connected
but not strongly connected (e.g.,

- **Strongly connected components (SCCs)** can be identified, but not every node is part of a nontrivial strongly connected component.





In-component: nodes that can reach the SCC,
Out-component: nodes that can be reached from the SCC.