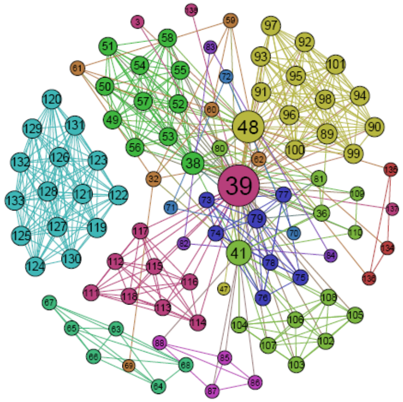


Social Network Analysis

Extracção de Conhecimento de Dados II

Márcia Oliveira
marcia@liaad.up.pt

João Gama
jgama@fep.up.pt

Outline

PART I

1. Background
2. Practical applications
3. Graph Theory:
    1. Types and representation of graphs
    2. Cliques
4. Fundamental concepts of SNA
5. Statistical measures to analyze networks
6. Link Analysis: hubs and authorities
    1. HITS algorithm
    2. PageRank algorithm
7. Properties of real-world networks
8. Community detection

Outline

PART II

1. Software for Social Network Analysis
2. Getting started with ***Gephi***: a practical exercise
    1. Extract your Facebook ego-network using netvizz application
    2. Import data to ***Gephi***
    3. Visualize, manipulate and analyze your own network
    4. Find communities and interpret them using your domain knowledge
3. Presentation of the *graph streaming* feature of ***Gephi***

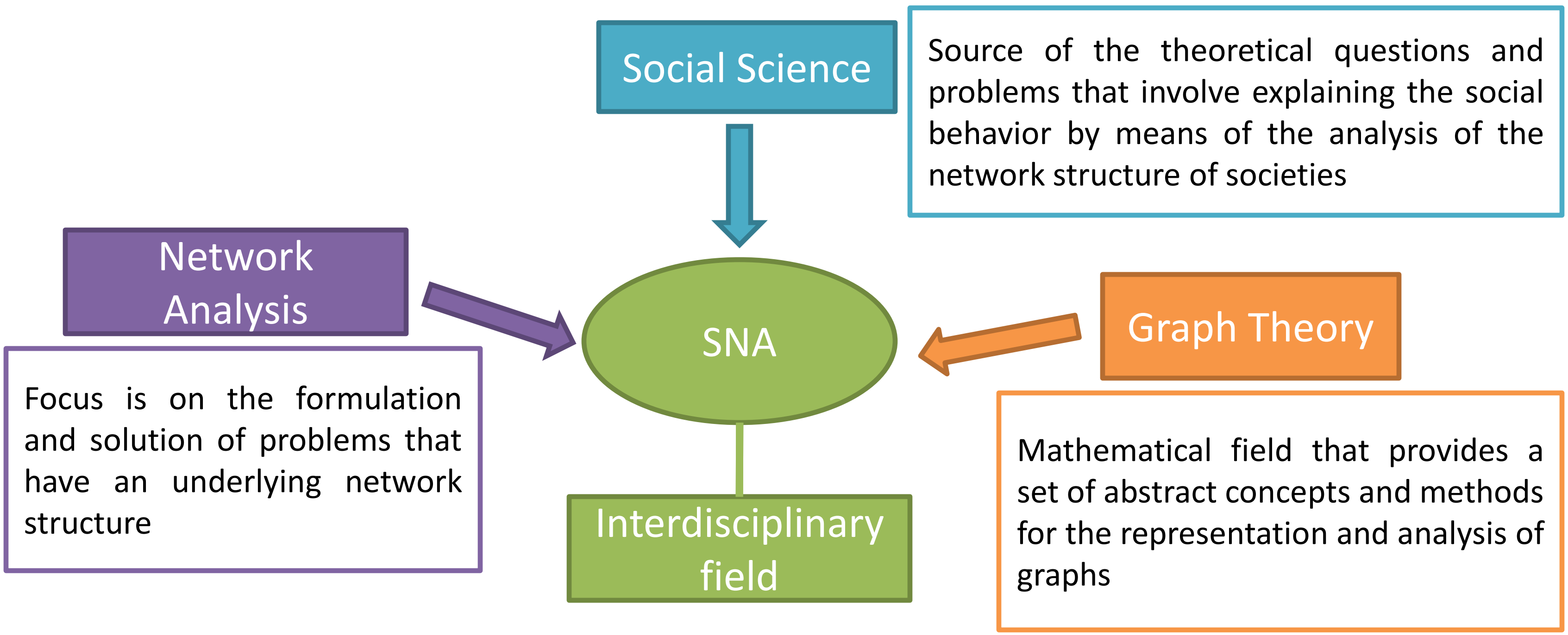
Part Ⅰ

Outline

1. **Background**
2. Practical applications
3. Graph Theory:
   1. Types and representation of graphs
   2. Cliques
4. Fundamental concepts of SNA
5. Statistical measures to analyze networks
6. Link Analysis: hubs and authorities
   1. HITS algorithm
   2. PageRank algorithm
7. Properties of real-world networks
8. Community detection

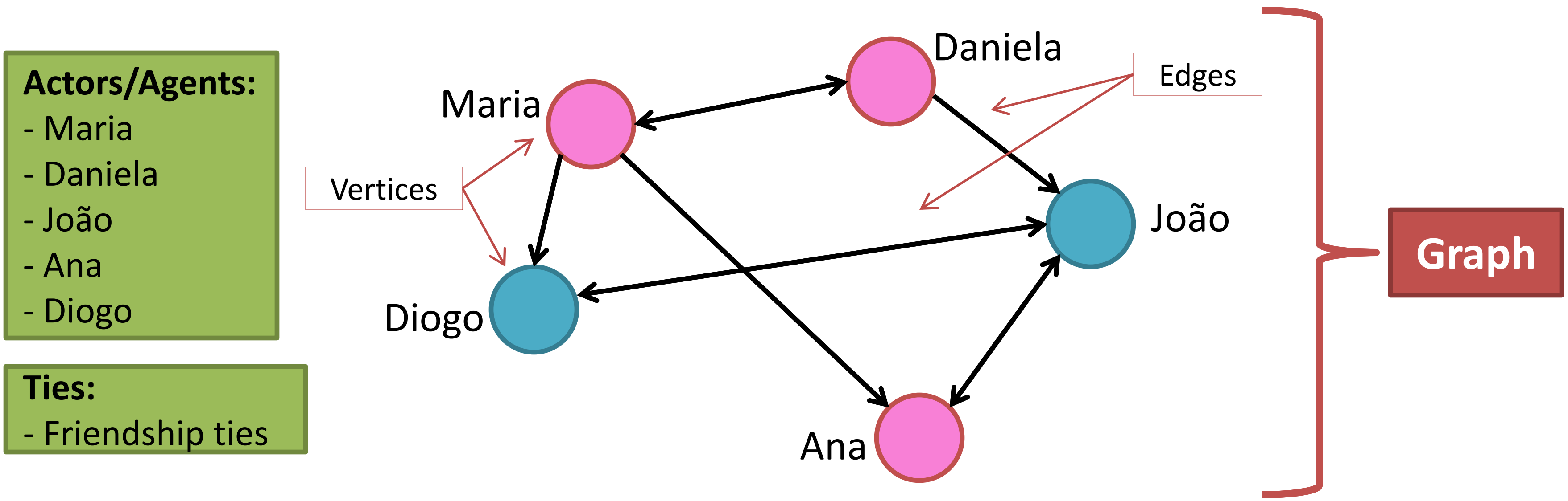
Background

- SNA - *Social Network Analysis* - focus on the relationships established between social entities (individuals, groups,etc.) rather in the social entities themselves
- SNA has its origins in both social science and in the broaden fields of Network Analysis and Graph Theory (Giorgos Cheliotis, 2010)



Background

Definition of social network (SN): a social network consists of a finite set(s) of actors and the relations (ties) defined on them (Wasserman and Faust, 1994).



❖The relationships can be of personal or profissional nature and can range from casual acquaintance to close familiar bonds

❖ Besides *social relations*, edges can also represent *flow of information/goods/money, interactions*, *similarities*, among others

❖ The structure of a SN is usually represented resorting to **graphs**.

Background

Terminology:

SNA is an interdisciplinary field with contributions from different knowledge areas; such variety of perspectives originated distinct terminology

Mathematics	Computer Science	Sociology	Physics
Vertex/Vertices	Node	Actor/Agent	Site
Edge	Link/Connection	Relational Tie	Bond

Background

Goal of SNA: examine both the <u>contents</u> and <u>patterns</u> of relationships in social networks in order to understand the relations among actors and the implications of these relationships. Common tasks of SNA involve the identification of:

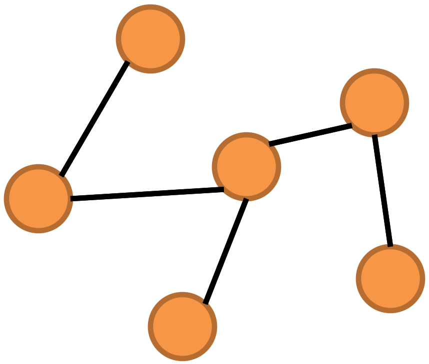
- Most central nodes
- Bridges, local bridges and gatekeepers
- Strong and weak ties
- Cliques
- Hubs and authorities
- Communities

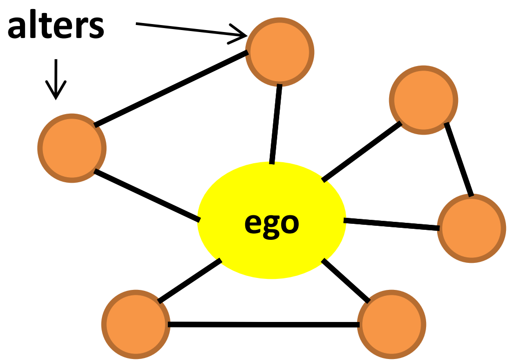
Background

Two main SNA approaches:

Sociocentric approach:

- **Focus on society as a whole**
- Gold standard approach
- <u>Assumption:</u> actors in a group interact more often than a random group of similar size
- <u>Goals:</u> measurement of the structural patterns of interactions between a defined number of actors and understanding of the impact of these patterns in outcomes





Egocentric approach:

- **Focus on actors**
- <u>Goal:</u> study of the relations surrounding a given individual – the *ego* or the *focal actor*
- Ego networks = Personal networks
- Networks are built and analyzed through the point of view of a single actor (the *ego*) and only the individuals that have connections to he/she (the *alters*) are represented

Outline

1. Background
2. **Practical applications**
3. Graph Theory:
    1. Types and representation of graphs
    2. Cliques
4. Fundamental concepts of SNA
5. Statistical measures to analyze networks
6. Link Analysis: hubs and authorities
    1. HITS algorithm
    2. PageRank algorithm
7. Properties of real-world networks
8. Community detection

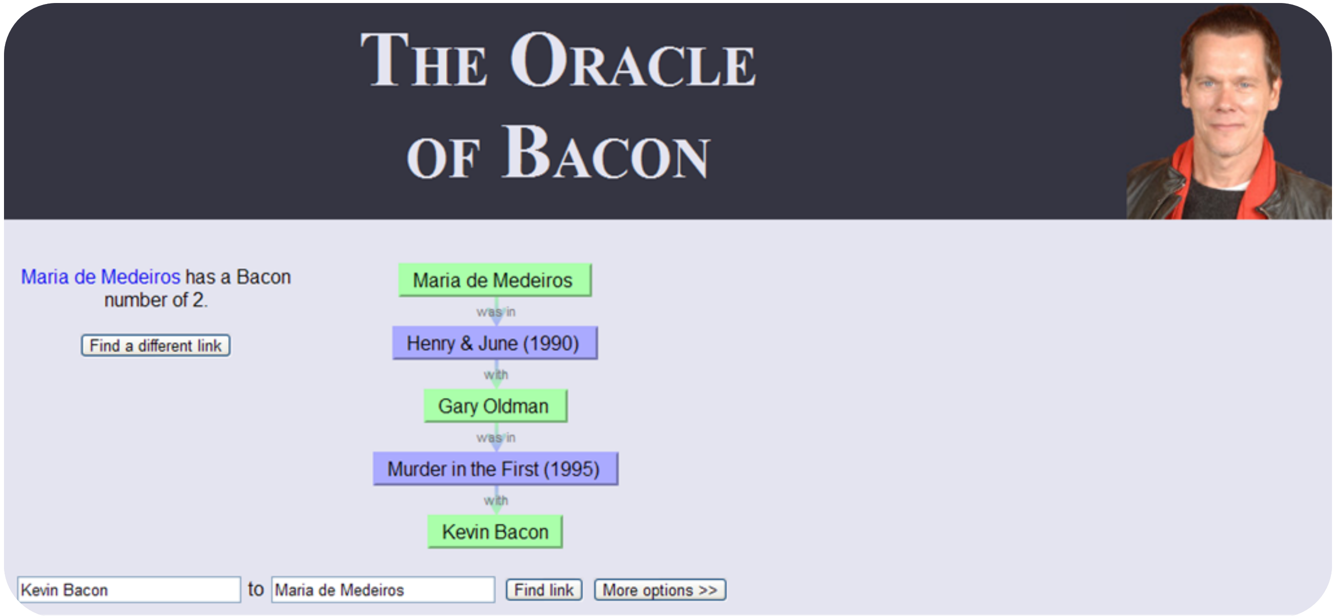
Practical Applications : social sciences

Collaboration networks:

→Networks of coappearance of actors in movies, in which two actors are connected if they appeared together in a movie

<u>Example:</u> The Oracle of Bacon http://oracleofbacon.org/

This *Oracle* uses the information available in IMDB and, based on the network of Kevin's Bacon coappearance, computes the shortest path from every actor/actress to Kevin Bacon.

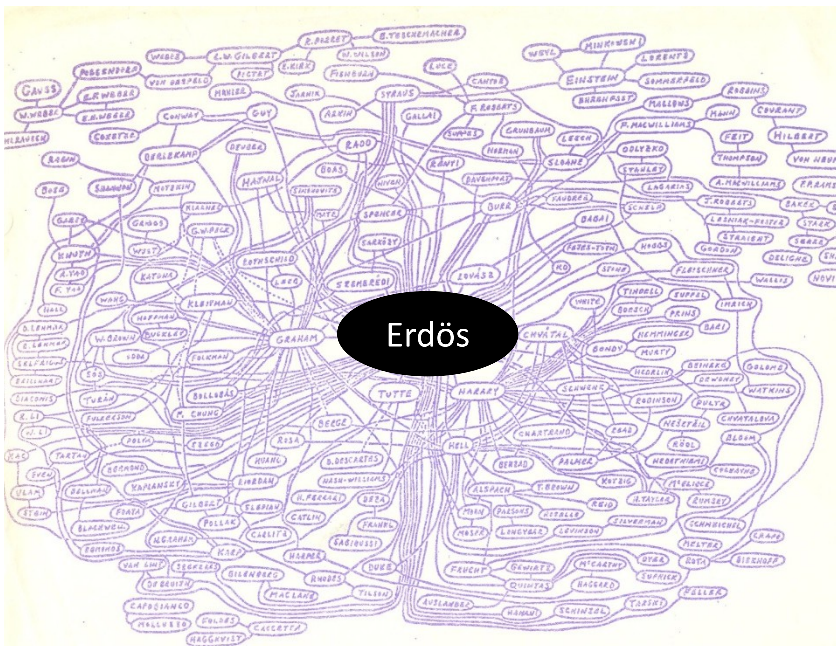


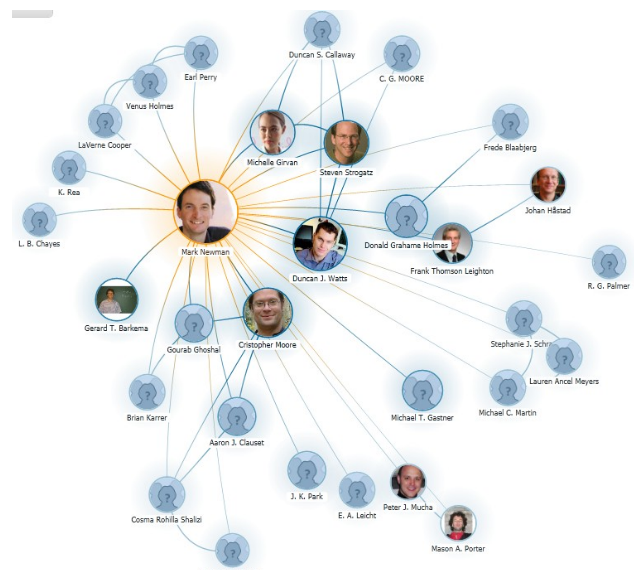
Practical Applications : social sciences

Collaboration networks:

→Networks of coauthorship among academics in which individuals are linked if they coauthored one or more papers *(scientific collaboration networks)*

<u>Example:</u> Paul Erdös coauthorship network; Microsoft Academic coauthorship network (http://academic.research.microsoft.com/)



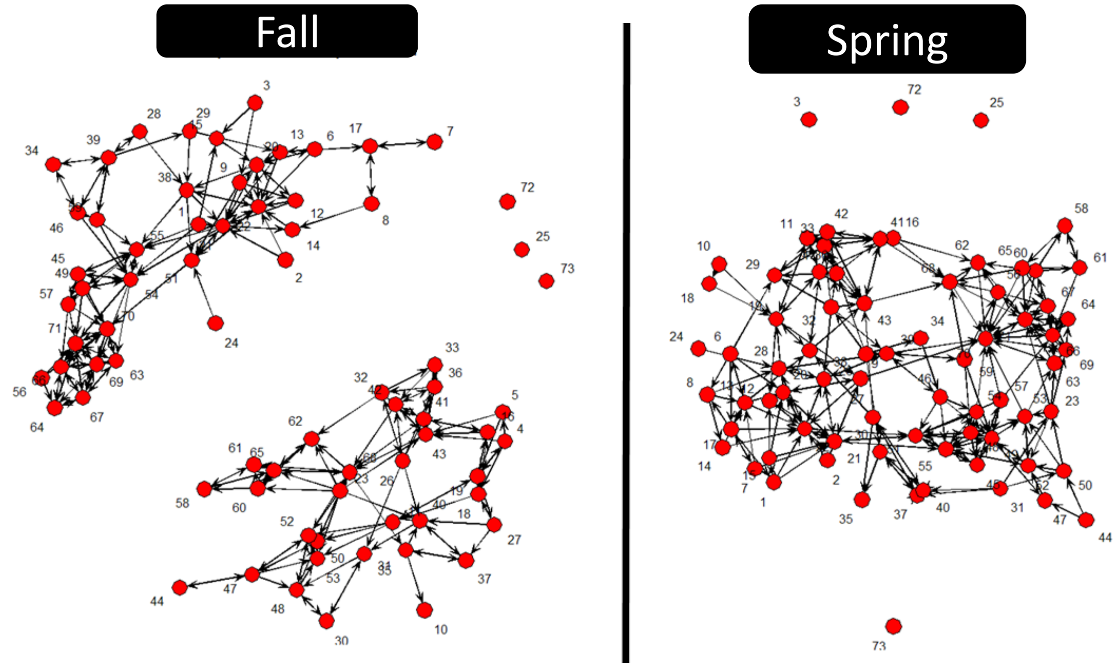


Friendship networks:

→Networks of friendship ties among high-school students

<u>Example:</u> Coleman data available in package `sna` of `R`

- Self-reported friendship ties among 73 boys in a small high-school in Illinois over the 1957-1958 academic year
- Both networks reflect answers to the question, “What fellows here in school do you go around with most often?”
- <u>Two networks:</u> Fall and Spring



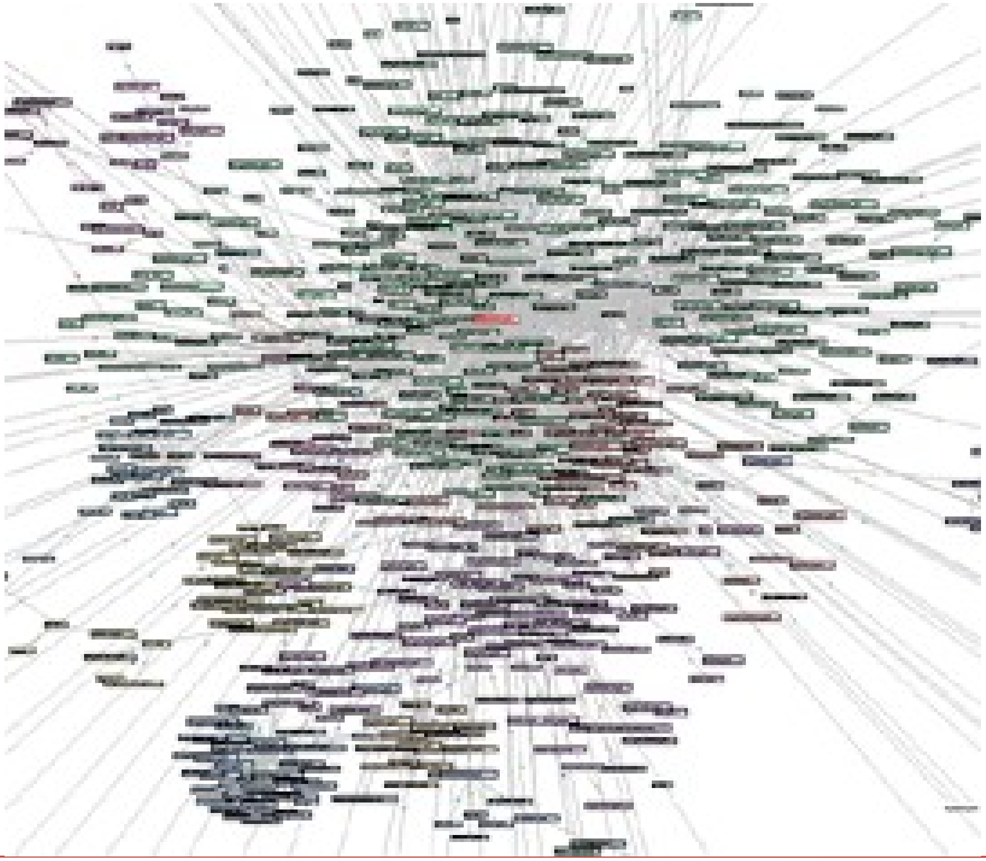
Practical Applications : social sciences

Communication networks:

→Networks of e-mail contacts between employees of a given company

<u>Example:</u> Enron e-mail corpus dataset

It contains real email data from about 150 users, mostly senior management of Enron company, organized into folders. The corpus contains a total of about 0.5M messages. This data was originally made public, and posted to the web, by the Federal Energy Regulatory Commission during its investigation of the accounting fraud of the company, known as the "Enron scandal".



Practical Applications : other domains

Life Sciences: use of network analysis to study *food chains* in different ecosystems

Network Operators (cable, mobile): use SNA-like methods to optimize the structure and capacity of their networks

Management: use SNA to analyze and improve the flow of communication within a given company, or between the company and their suppliers/clients; study of the diffusion of innovation within industrial clusters

Army: use SNA to identify criminal and terrorist networks from traces of collected communications and identify key players in these networks

Health: use SNA in the study of the spread of contagious diseases, such as HIV, through the analysis of networks of sexual contacts

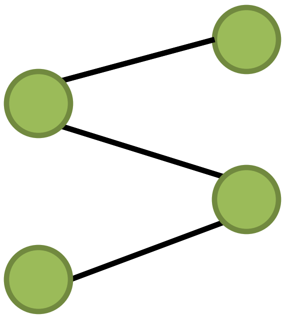
(Giorgos Cheliotis, 2010)

Outline

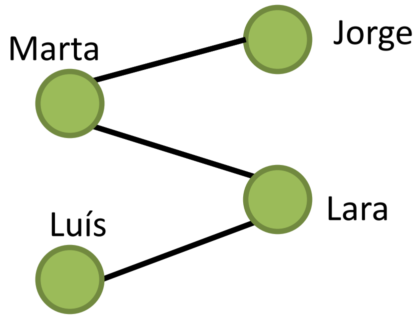
1. Background
2. Practical applications
3. **Graph Theory:**
    1. **Types and representation of graphs**
    2. **Cliques**
4. Fundamental concepts of SNA
5. Statistical measures to analyze networks
6. Link Analysis: hubs and authorities
    1. HITS algorithm
    2. PageRank algorithm
7. Properties of real-world networks
8. Community detection

Graph Theory
1. Types and representation of graphs

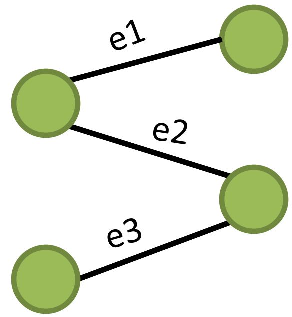
Unlabeled



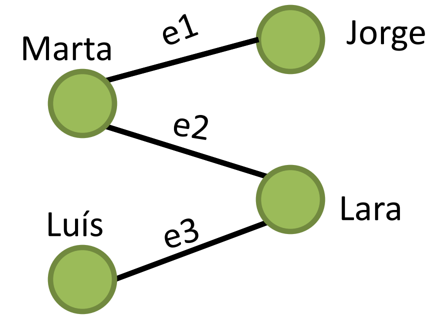
Vertex-labeled



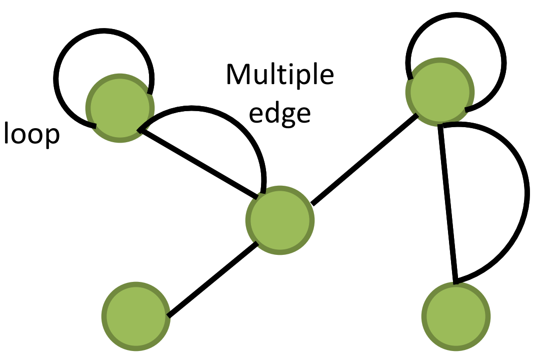
Edge-labeled



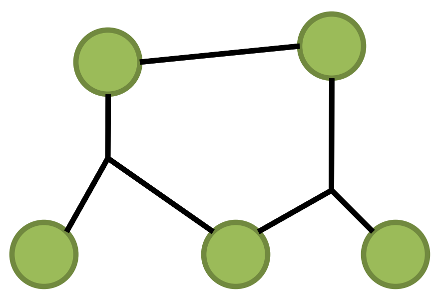
Vertex and Edge-labeled



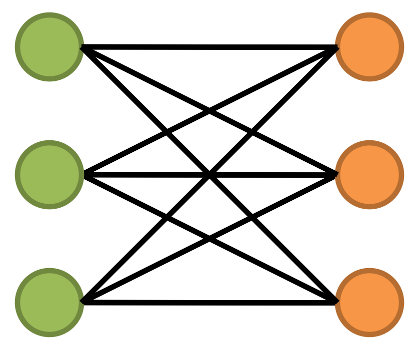
Multigraphs



Hypergraphs



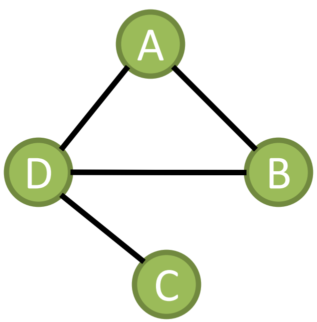
Bipartite Graphs



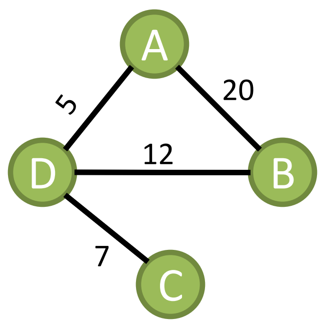
Graph Theory
1. Types and representation of graphs

Undirected graphs

Unweighted



Weighted



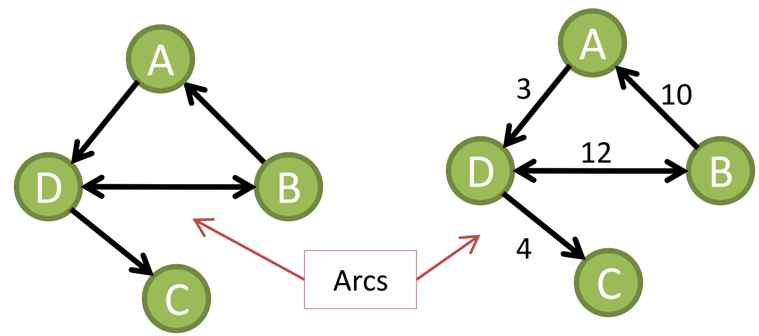
Graph G(V,E)

V(G)={A,B,C,D} ▶ **Vertex set**

E(G)={(A,B),(A,D),(B,D),(D,C)} ▶ **Edge set**

Directed graphs = Digraphs

Unweighted **Weighted**



❖ If only one direction is allowed then the graph is <u>*oriented*</u> instead of <u>*directed*</u>

Digraph D(V,A)

V(D)={A,B,C,D} ▶ **Vertex set**

A(D)={(A,D), (D,C), (B,A),(B,D),(D,B) } ▶ **Edge set**

Graph representation schemes

List structures

- ❖ Incidence lists
- ❖ **Adjacency lists**

Appropriate to store *sparse* graphs

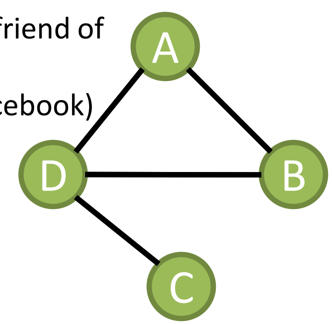
Matrix structures

- ❖ Incidence matrices $G_{n.m}$
- ❖ **Adjacency matrices $G_{n.n}$**
- ❖ Laplacian matrices
- ❖ Distance matrices

Appropriate to represent *full* matrices

Undirected graphs

Who is friend of whom?
(e.g. Facebook)



Adjacency List

Vertex	Vertex
A	B
A	D
B	D
C	D

Adjacency Matrix

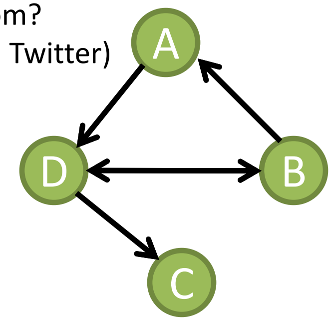
Vertex	A	B	C	D
A	-	1	0	1
B	1	-	0	1
C	0	0	-	1
D	1	1	1	-

- Symmetric
- Binary

Graph Theory
1. Types and representation of graphs

Directed graphs

Who follows whom?
(e.g. Twitter)



Adjacency List

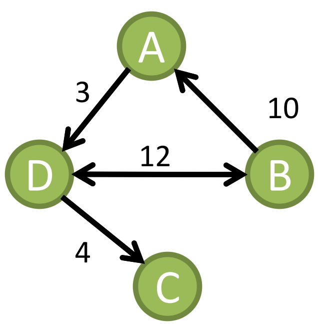
Vertex	Vertex
A	D
B	A
B	D
D	B
D	C

Adjacency Matrix

Vertex	A	B	C	D
A	-	0	0	1
B	1	-	0	1
C	0	0	-	0
D	0	1	1	-

- Non-symmetric
- Binary

Weighted graphs



Adjacency List

Vertex	Vertex	Weight
A	D	3
B	A	10
B	D	12
D	B	12
D	C	4

Adjacency Matrix

Vertex	A	B	C	D
A	-	0	0	3
B	10	-	0	12
C	0	0	-	0
D	0	12	4	-

Graph Theory
2. Cliques

Clique:

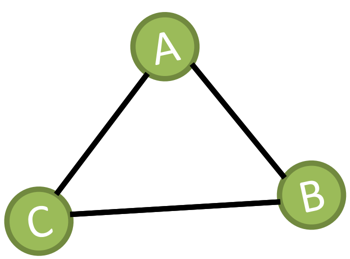
❖ Cliques are complete subgraphs of a given size

❖ In other words, clique is a subset of the vertices of a graph such that every pair of vertices in the subset is connected by an edge or, in other words, all vertices are neighbors (or adjacent);

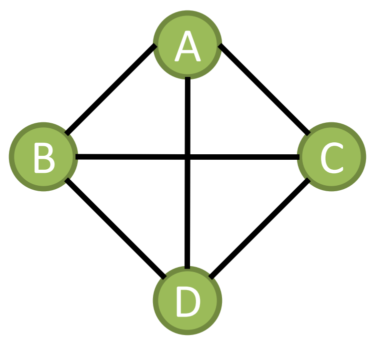
❖ The opposite of a clique is a *coclique*, or *independent set*

❖ In the *Social Networks* context, cliques can be understood as a group of people all of whom knows each other.

K_3 - Clique of size 3 (tryad)



K_4 - Clique of size 4



Maximum clique: clique with the largest possible size

Clique number: number of vertices in the maximum clique

In **large graphs**, the problem of finding all cliques, or the maximum clique, is a NP-complete computational problem.

Outline

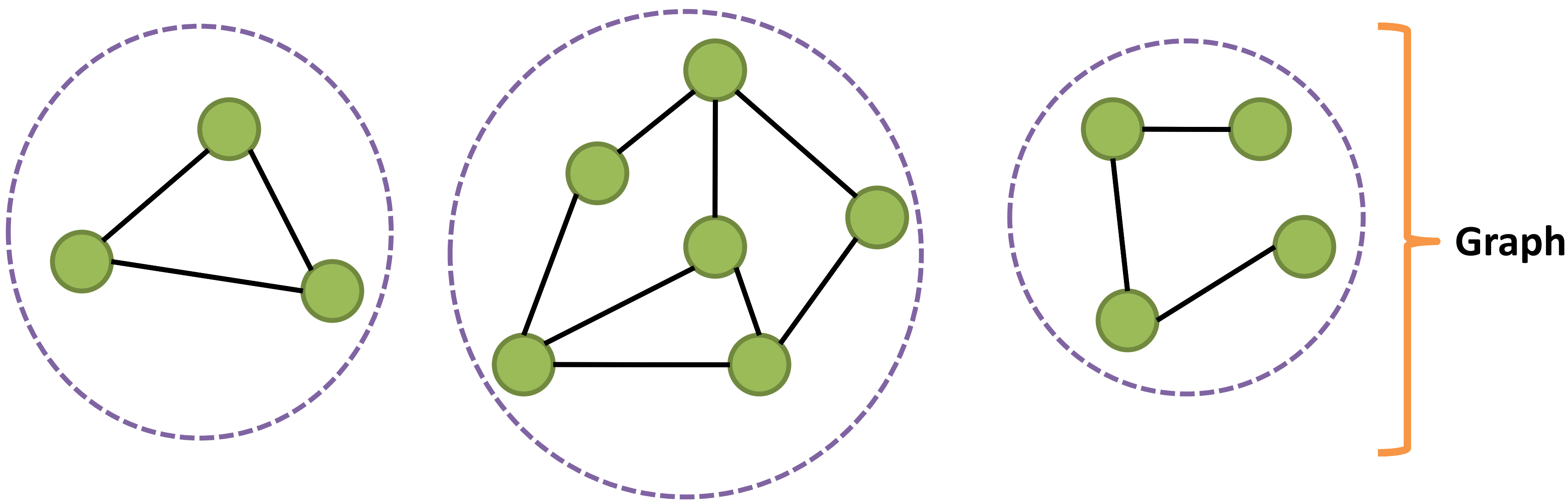
1. Background
2. Practical applications
3. Graph Theory:
    1. Types and representation of graphs
    2. Cliques
4. **Fundamental concepts of SNA**
5. Statistical measures to analyze networks
6. Link Analysis: hubs and authorities
    1. HITS algorithm
    2. PageRank algorithm
7. Properties of real-world networks
8. Community detection

Fundamental Concepts of SNA

Connected component:

❖ <u>Maximal connected subgraph</u> or <u>densely connected subgraph</u> (Easley and Kleinberg, 2010) is a:

- ❖ Connected subgraph: for any pair of vertices there is, at least, one path going from one vertex to another
- ❖ And is maximal: adding a vertex, the subgraph is no more connected
- ❖ The subgraph is a free-standing piece of the graph, not a connected part of a larger piece



Fundamental Concepts of SNA

Bridge (or cut-edge/cutpoint):

❖ Edge connecting two vertices that would belong to different connected components, or regions, in a graph if this edge was deleted

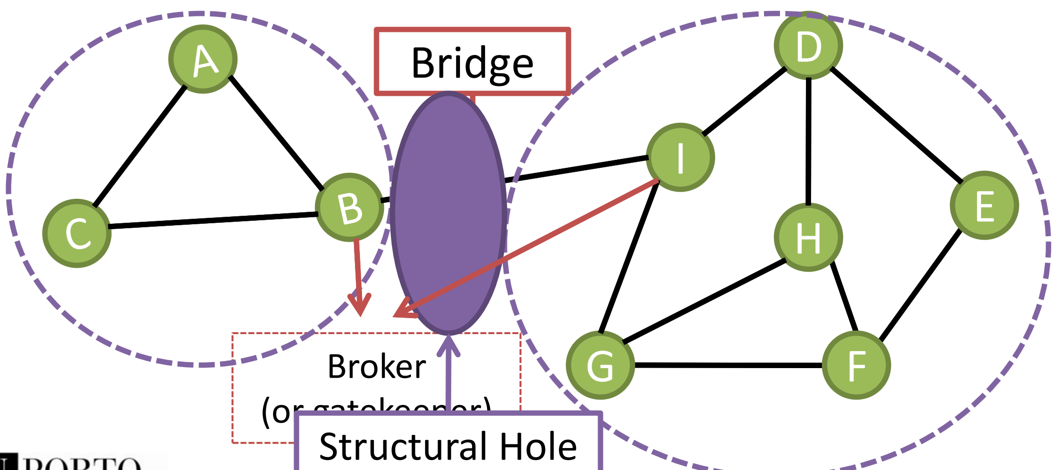
❖ <u>In the *Social Networks* context</u>, bridges can be understood as connections outside an individual's circle of acquaintances; the endpoints of a bridge are commonly called ***brokers***, or ***gatekeepers***, in SNA

❖ Usually, bridges are associated to <u>weak ties</u> (though not every weak tie is a bridge)

Advantages :

- ✓ Eases the communication between groups
- ✓ Promotes the spread of innovation
- ✓ Access to new information and resources

Bridge ***b=(B,I)*** links two regions of the network:
R1={A,B,C}
R2={D,E,F,G,H,I}



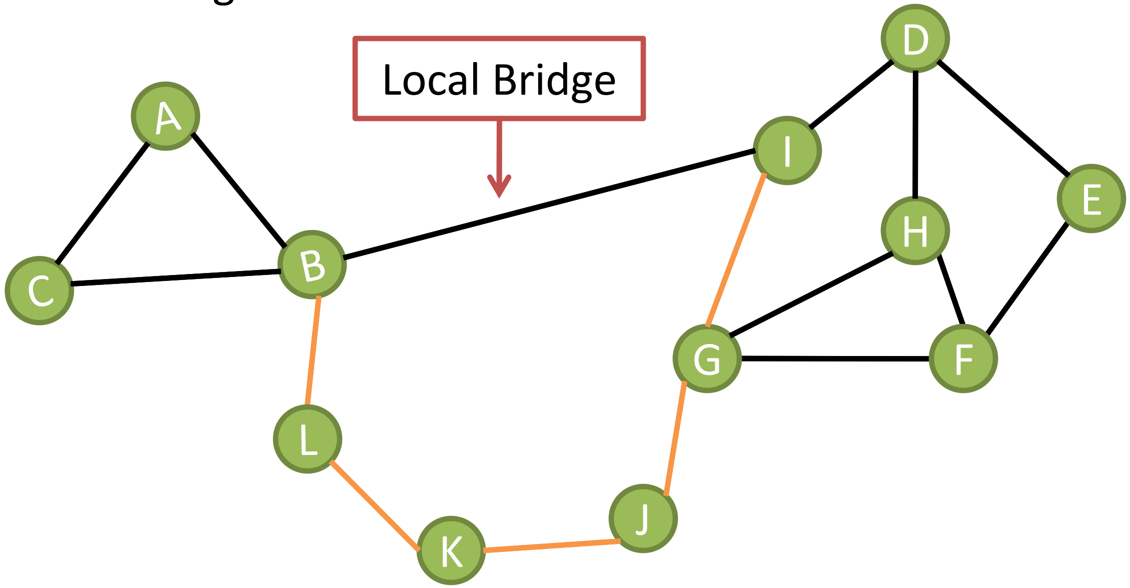
If the bridge was removed the network would be replaced by two connected components, creating a **<u>structural hole</u>** that prevents the communication between them.

Fundamental Concepts of SNA

Local bridge:

❖ Edge joining two vertices that have no direct neighbors (or adjacent vertices) in common, which means that if we remove this edge the *geodesic distance* between those two vertices will increase

❖ A ***local bridge*** is a link that reduces drastically the distance between two sets of actors, though it does not define an unique path from actors of one region to actors in another region



Local bridge ***lb=(B,I)*** shortens the distance between the set of actors *{A,B,C}* and the set of actors *{D,E,F,G,H,I}*.

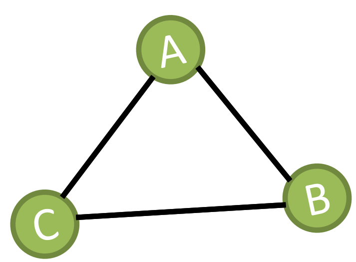
The removal of the *local bridge* would not hinder the access to certain parts of the network, since there is always an alternative path, e.g. *{B,L,K,J,G,I}*, though it is longer.

Fundamental Concepts of SNA

Principle of transitivity: if two people in a social network have a friend in common, then there is a heightened probability that they will become friends themselves at some point in the future. Transitivity is, therefore, a property of ties. (Rapoport, 1953)

- ❖ In the *Social Networks* parlance, it means that a friend of your friend is also likely to be your friend.
- ❖ The increase of the linking probability is usually motivated by *opportunity*, *trusting* and *incentive*
- ❖ Graphically, this phenomenon is represented by the *closure of the third side of the triangle*, forming a K_3 clique (clique of size 3). This is known as **triadic closure.**
- ❖ The concept of *triadic closure* only acquires significance when the same network is analyzed over time

Time point t_{i+1}



Strong ties are more often transitive than **weak ties**

Fundamental Concepts of SNA

Homophily:

❖ Homophily can be defined as the tendency of people to establish relationships with people sharing similar characteristics (e.g. age, gender, class, status, beliefs etc.).

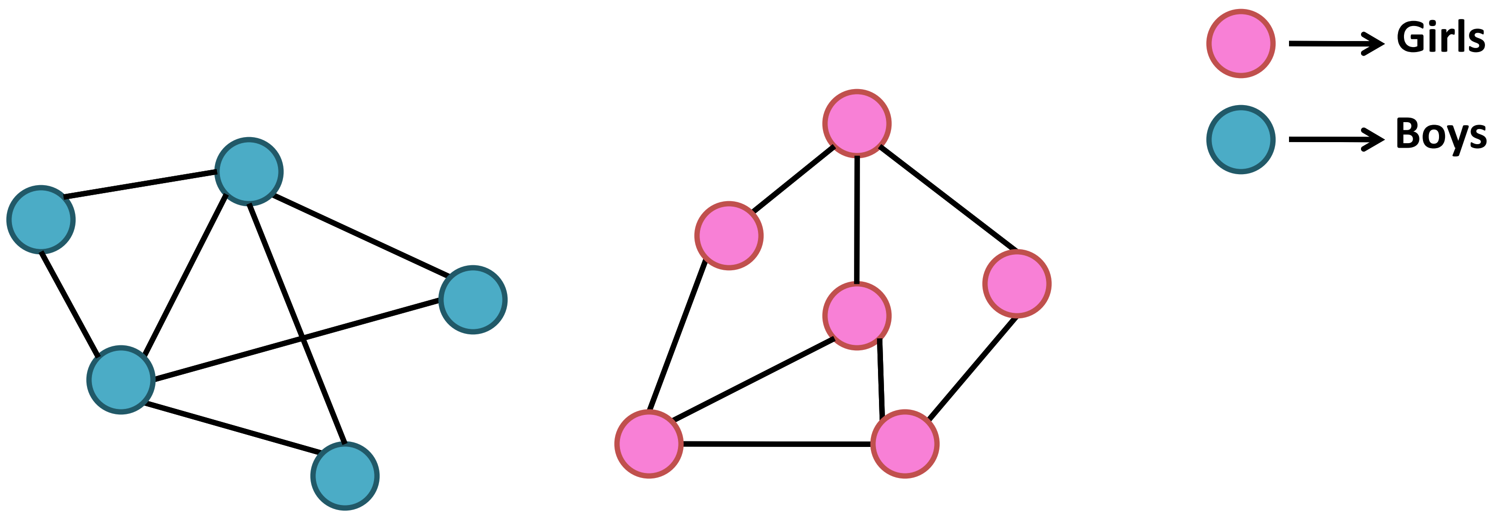
❖ The opposite of *homophily* is *heterophily*

Advantage :

- ✓ Facilitates communication and the creation of bonds

Drawback:

- ✓ Hampers the generation of new ideas and the adoption of innovation inside the group

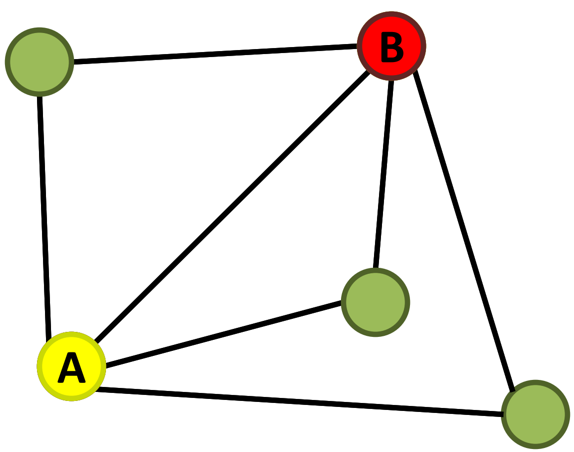


Fundamental Concepts of SNA

Structural Equivalence:

❖ Mathematical property that expresses the similarity between actors in a social network based on the neighbors they share (or, equivalently, the number of identical ties they have)

❖ Two actors are <u>structurally equivalent</u> if their positions can be swapped, without modifying the overall structure of the network



The ***approximate*** **structural equivalence** can be used as a basis for <u>hierarchical clustering</u> of social networks

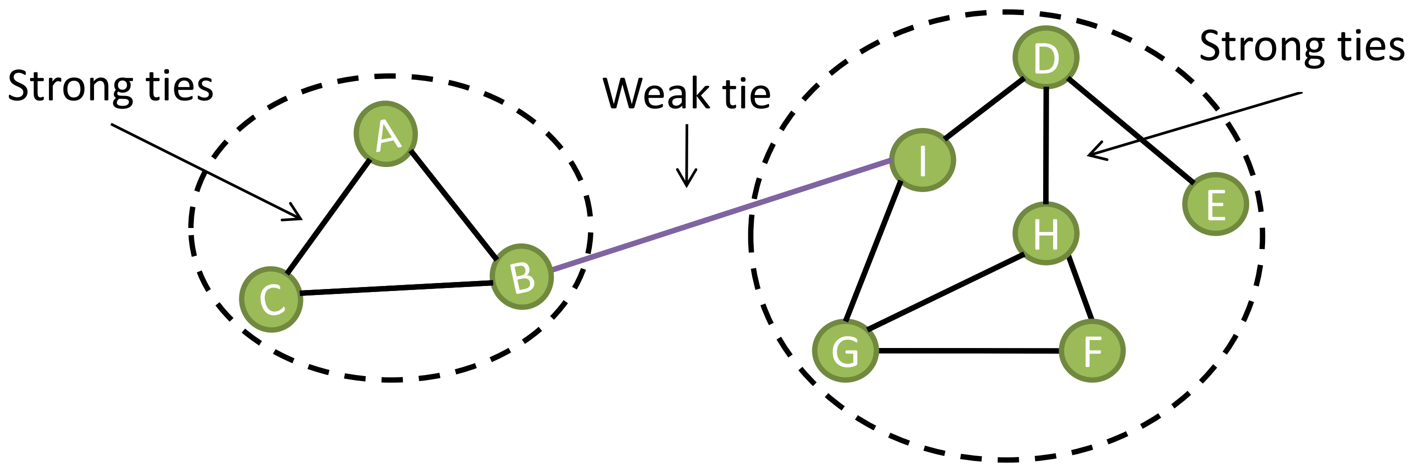
Fundamental Concepts of SNA

Strong and Weak ties:

In *friendship networks*, there are usually two types of relations:

✓ **Close friendship:** originates densely knit group of individuals ➡ **Strong ties**

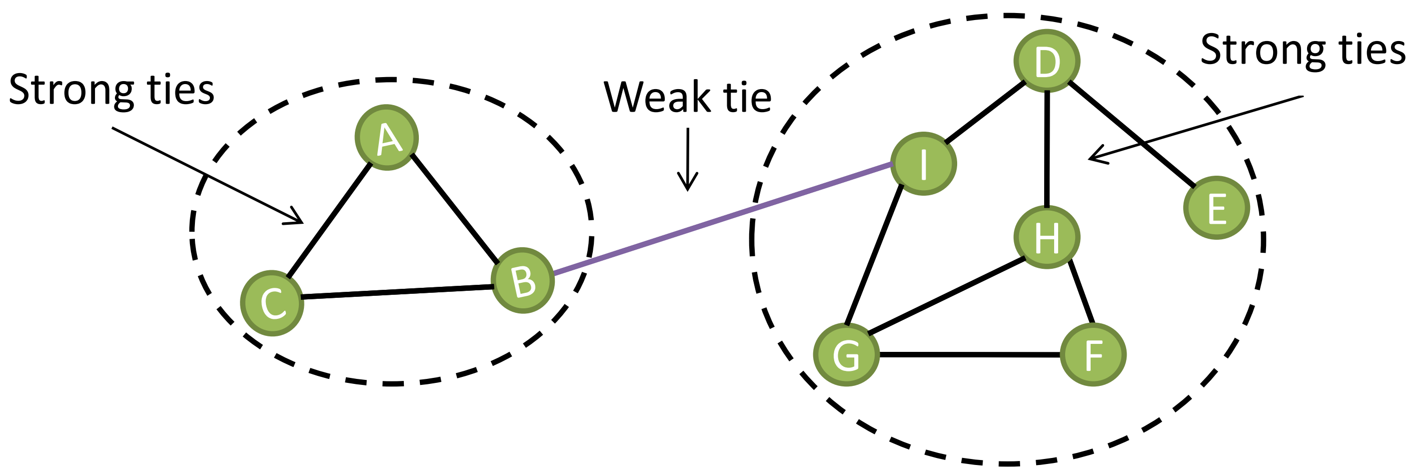
✓ **Acquaintance:** these relationships usually act as ***bridges*** between different connected components ➡ **Weak ties**



Fundamental Concepts of SNA

The Strength of the Weak Ties: (Granovetter, 1973)

- **Hypothesis:**
  Useful information is normally achieved through **weak ties** over **strong** ones
- **Findings of Granovetter's research:**
  Weak ties enable reaching crucial information, such as good job opportunities, which are not accessible via strong ties.



Outline

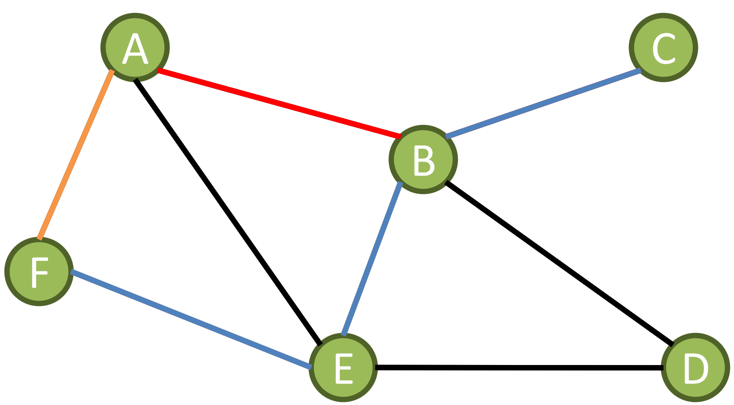
1. Background
2. Practical applications
3. Graph Theory:
    1. Types and representation of graphs
    2. Cliques
4. Fundamental concepts of SNA
5. **Statistical measures to analyze networks**
6. Link Analysis: hubs and authorities
    1. HITS algorithm
    2. PageRank algorithm
7. Properties of real-world networks
8. Community detection

Statistical Measures to Analyze Networks: Basic Concepts

Path: sequence of nodes in which consecutive pairs of (non-repeating) nodes are linked by an edge; the first vertex of a path is called the *start vertex* and the last vertex of the path is called the *end vertex*

Geodesic distance: the geodesic distance between two nodes/vertices is given by the number of edges connecting them in the shortest path; length of the shortest path

Eccentricity of a vertex: greatest geodesic distance between a given vertex *v* and any other in the graph



Path

Eccentricity of vertex C

Greatest shortest path= {(C,B),(B,E),(E,F)}
e(C)=3

$$\epsilon_v = \max_{i \in V(G) \setminus v} d(v,i)$$

dC,A=2

Statistical Measures to Analyze Networks

Actor-Level measures:

Measures of Centrality: *centrality* is a general measure of how the position of a vertex is within the overall structure of the graph; it helps identify the key players in the network. The best known are:

- **Degree, Valency or Prestige**
  - **In-degree**
  - **Out-degree**
- **Betweenness**
- **Closeness**
- **Eigenvector centrality**

Network-Level measures: to assess the overall structure of the network

- **Diameter/Radius**
- **Average degree**
- **Density**
- **Average geodesic distance**
- **Reciprocity**
- **Clustering**

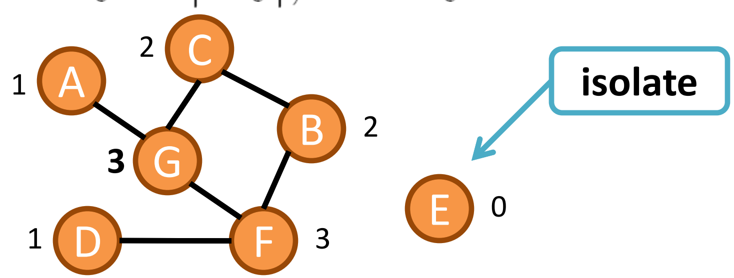
Statistical Measures to Analyze Networks: Actor-level measures

Degree, Valency or Prestige

- Number of neighbors (or connections) of a vertex *v*
- It is a measure of the number of individuals in the network that a specific actor can reach or, alternatively, it is a measure of the involvement of the actor in the network

Undirected networks:

$$k_v = |N_v|, 0 < k_v < n$$



Weighted networks:

$$k_v^w = \sum_{u \in N_v} w_{vu}$$

Directed networks:

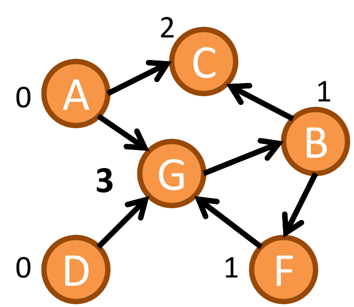
In-degree
- Number of incoming vertices
- Measure of support

k_v^+

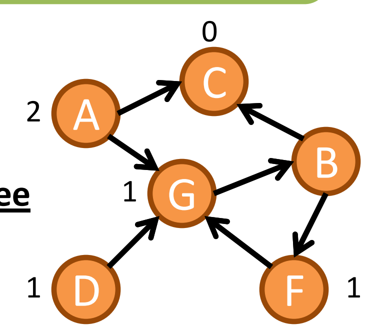
Out-degree
- Number of outgoing vertices
- Measure of influence

k_v^-

<u>In-degree</u>



<u>Out-degree</u>



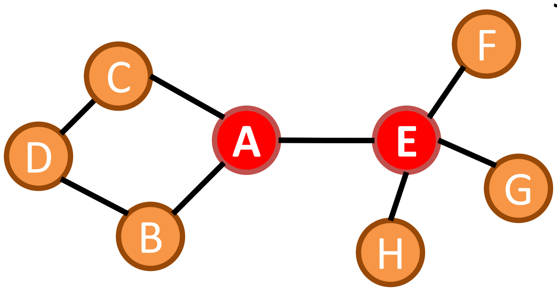
Statistical Measures to Analyze Networks: Actor-level measures

Betweenness

- The extent to which a node lies between other nodes in the network
- Vertices with high betweenness occupy critical roles in the network structure, since they usually have a network position that allow them to work as an interface between tightly-knit groups, being "vital" elements in the connection between different regions of the network.
- In the *Social Networks* context, these nodes are known as the **gatekeepers**

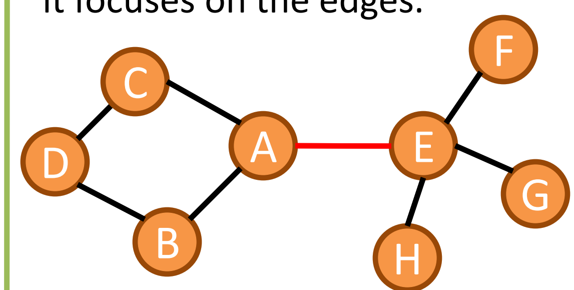
$$b_v = \sum_{s,t \in V(G) \setminus v} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

Total number of shortest paths between vertices ***s*** and ***t***



Edge betweenness:

The idea is the same but instead of focusing on nodes, it focuses on the edges.



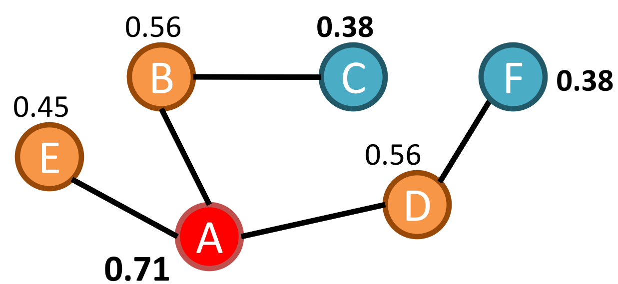
$$b_e = \sum_{u,v \in V(G)} \frac{\sigma_{uv}(e)}{\sigma_{uv}}$$

Statistical Measures to Analyze Networks: Actor-level measures

Closeness

- Mean length of all shortest paths from one node to all other nodes in the network
- Only computed for vertices within the largest component of the network
- Measure of reachability that gives an idea about how long it will take to reach other nodes from a given starting node
- In the *Social Network* context, closeness measures how fast can a given actor reach everyone in the network

$$Cl_v = \frac{n-1}{\sum_{u \in V(G) \setminus v} d(u,v)}$$



Node A has the highest closeness to all other nodes in the network.

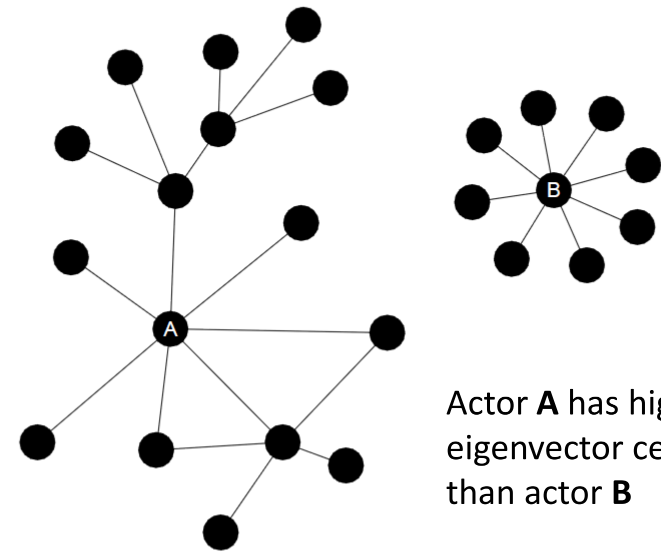
Nodes C and F are the ones with lowest closeness in the network.

Statistical Measures to Analyze Networks: Actor-level measures

Eigenvector Centrality

- This metric is based on the idea that the *power* and *status* of an actor is recursively defined by the *power* and *status* of his/her alters (or neighbors)
- The eigenvector of a node is proportional to the sum of the eigenvector centralities of all its direct neighbors
- It measures how well a given actor is connected to other well-connected actors

Degree VS Eigenvector centrality: **eigenvector centrality** is a more elaborated version of the **degree**, once it assumes that not all connections have the same importance by taking into account not only the quantity, but especially the quality of these connections.



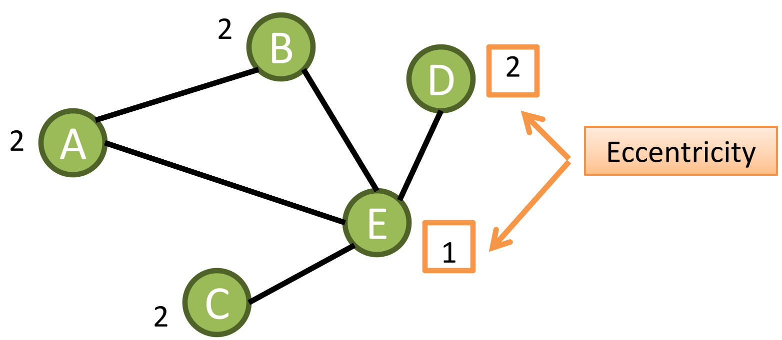
Actor **A** has higher eigenvector centrality than actor **B**

Statistical Measures to Analyze Networks: Network-level measures

Diameter/Radius

Diameter:
- Longest shortest path between any two nodes in the network
- Maximum eccentricity of the set of vertices in the network
- Sparser networks have generally greater diameter

Radius:
- Minimum eccentricity of the set of vertices in the network



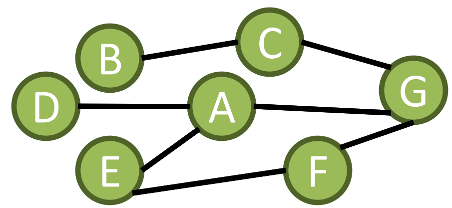
Diameter= 2

Radius = 1

Average Geodesic Distance

- The <u>average geodesic distance</u> gives an idea of how far apart nodes will be, on average, in the network
- Measures the efficiency of information flow within the network

$$l = \frac{1}{\frac{1}{2}n(n+1)} \sum_{i \geq j} d_{ij}$$

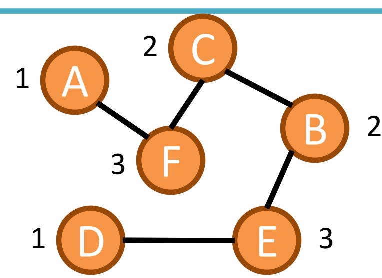


$$l = \frac{3+2+1+1+2+1+1+4+4+3+2+3+3+2+1+2+3+2}{\frac{1}{2}7(7+1)} = \frac{40}{28} \approx 1.43$$

Average Degree

- Measure of the overall connectivity of the network
- Computed as the mean of the degrees of all network's vertices

$$\overline{k} = \frac{1}{n} \sum_{i=1}^{n} k_i$$



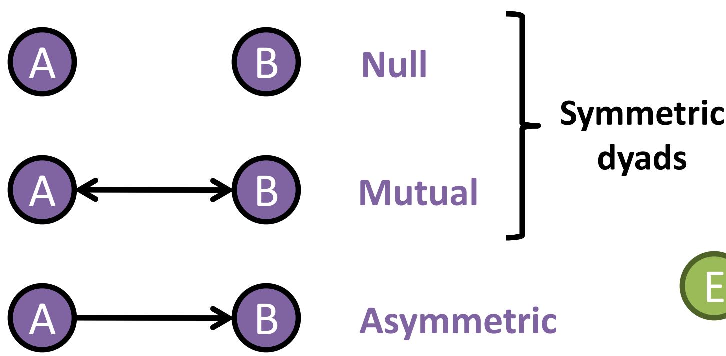
$$\overline{k} = \frac{1+3+2+2+3+1}{6} = \frac{12}{6} = 2$$

Statistical Measures to Analyze Networks: Network-level measures

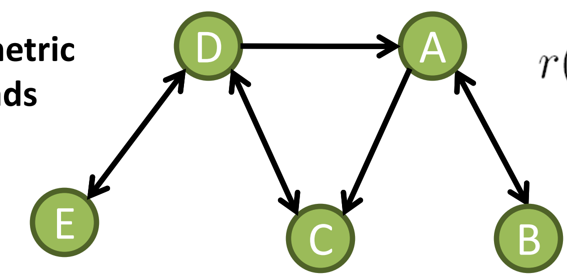
Reciprocity

- Measures the tendency of pair of vertices to form reciprocal connections between each other
- Specific metric for directed networks
- *Reciprocity* is computed as the proportion of symmetric dyads in a given digraph and its value represents the probability that two vertices share the same type of connection.

Classes of isomorphism for dyads



$$r(D) = \frac{s(D)}{C_2^n} = \frac{mut(D) + null(D)}{C_2^n}, 0 < r < 1$$



$$r(D) = \frac{3+5}{C_2^5} = \frac{8}{10} = 0.8$$

Statistical Measures to Analyze Networks: Network-level measures

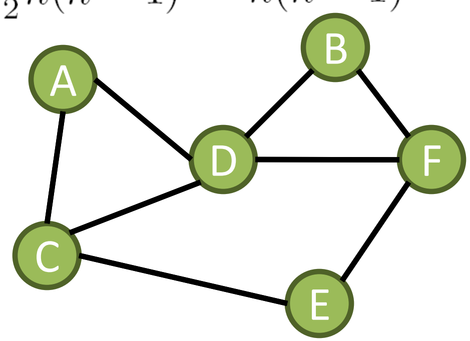
Density

- The ***density*** indicates the level of connectedness in a network, with high values being associated to **dense networks** and low values associated to **sparse networks**
- Density is simply the proportion of edges/arcs (*m*) in the graph/digraph relative to the maximum possible number of edges/arcs (m_{max})
- A perfectly connected network is called a ***clique***, or ***complete graph***, and has a maximum density of 1

Undirected networks:

$$\rho(G) = \frac{m}{m_{max}} = \frac{m}{\frac{1}{2}n(n-1)} = \frac{2m}{n(n-1)},\ 0 < \rho < 1$$

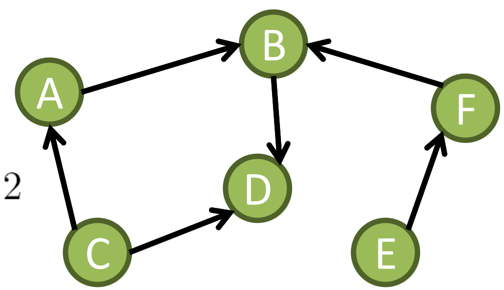
$$\rho(G) = \frac{2\times 8}{6(6-1)} \approx 0.53$$



Directed networks:

$$\rho(D) = \frac{m}{m_{max}} = \frac{m}{n(n-1)}$$

$$\rho(D) = \frac{6}{6(6-1)} = 0.2$$

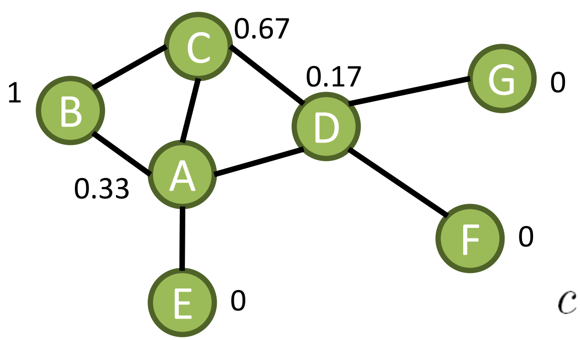


Statistical Measures to Analyze Networks: Network-level measures

Transitivity or Clustering

- *Transitivity*, or *clustering*, is a property that considers the density/cohesion of a node's neighborhood, in its **local version**; or the density of triangles in a network, in its **global version**.
- This property is quantified by computing a ***clustering coefficient***, that can be local (computed for each node) or global (computed for the whole network)
  - ✓ **Local clustering coefficient:** fraction of pairs of vertices, that are neighbors of a given vertex *v*, that are connected to each other by edges.
  - ✓ **Global clustering coefficient:** average of all local coefficients
- Clustering is useful once it indicates the presence of sub-communities in the network

Local clustering coefficients



$$\left.c_i = \frac{2|e_{jk}|}{k_i(k_i-1)} : v_j, v_k \in N_i, e_{jk} \in E \right\} \textbf{Local}$$

Global clustering coefficient

$$c = \frac{1+0.67+0.33+0.17}{7} = 0.31$$

$$\left.c = \frac{1}{n}\sum_v c_v \right\} \textbf{Global}$$

Outline

1. Background
2. Practical applications
3. Graph Theory:
    1. Types and representation of graphs
    2. Cliques
4. Fundamental concepts of SNA
5. Statistical measures to analyze networks
6. **Link Analysis: hubs and authorities**
    1. **HITS algorithm**
    2. **PageRank algorithm**
7. Properties of real-world networks
8. Community detection

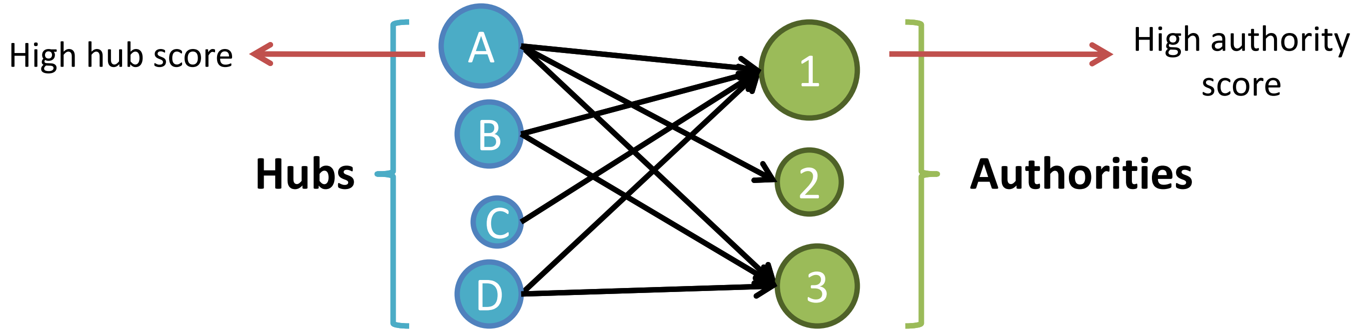
Link analysis: hubs and authorities

Authorities

- Authorities are web pages cited by many different ***hubs***
- Good ***authorative pages*** are reliable sources of information about a given topic
- Using the network parlance, ***authorities*** are nodes receiving many inward links
- The relevance of an authority is “measured” by the number of inward links

Hubs

- Hub can be understood as a web page that points to many other web pages or, in other words, as a compilation of web pages that address a specific topic
- Using the network parlance, ***hub*** is a node with many outward links
- A good hub is a site that points to good ***authorative sites***



Link analysis: hubs and authorities
1. HITS algorithm

HITS algorithm

- **HITS** (*Hypertext Induced Topic Selection*) is a link analysis algorithm developed by Kleinberg (1999)
- HITS computes and returns 2 scores, for each node *v* in the network: the **authority score** *auth(v)* and the **hub score** *hub(v)*
- These scores provide information about the potential of each node to be an *authority* or a *hub*, thus indicating how valuable is the information carried by it

Link analysis: hubs and authorities
1. HITS algorithm

HITS algorithm

HITS is an iterative algorithm based on the repeated update of two basic rules:

- **Authority Update Rule:** for each page *p* (or node *v*), update ***auth(p)*** to be the sum of the hub scores of all pages that point to it. A high authority score is assigned to *p* if it is linked to pages that are recognized as important hubs of information.

- **Hub Update Rule:** for each page *p* (or node *v*), update ***hub(p)*** to be the sum of the authority scores of all pages that it points to. A high hub score is assigned to *p* if it links to nodes that are considered to be authorities in a given topic.

Link analysis: hubs and authorities
1. HITS algorithm

HITS algorithm

- Given a query (topic) ***Q***, collect the root set of pages **S=**{$s_1, s_2, s_3, \dots, s_n$}
- Set **S** is expanded to set **T= S** U {d|s->d or d->s, s ϵ **S**}
- **Initialization:** start with *auth(p) = 1* and *hub(p) = 1*, for every *p* ϵ **T,** and choose

the number of iterations *k*

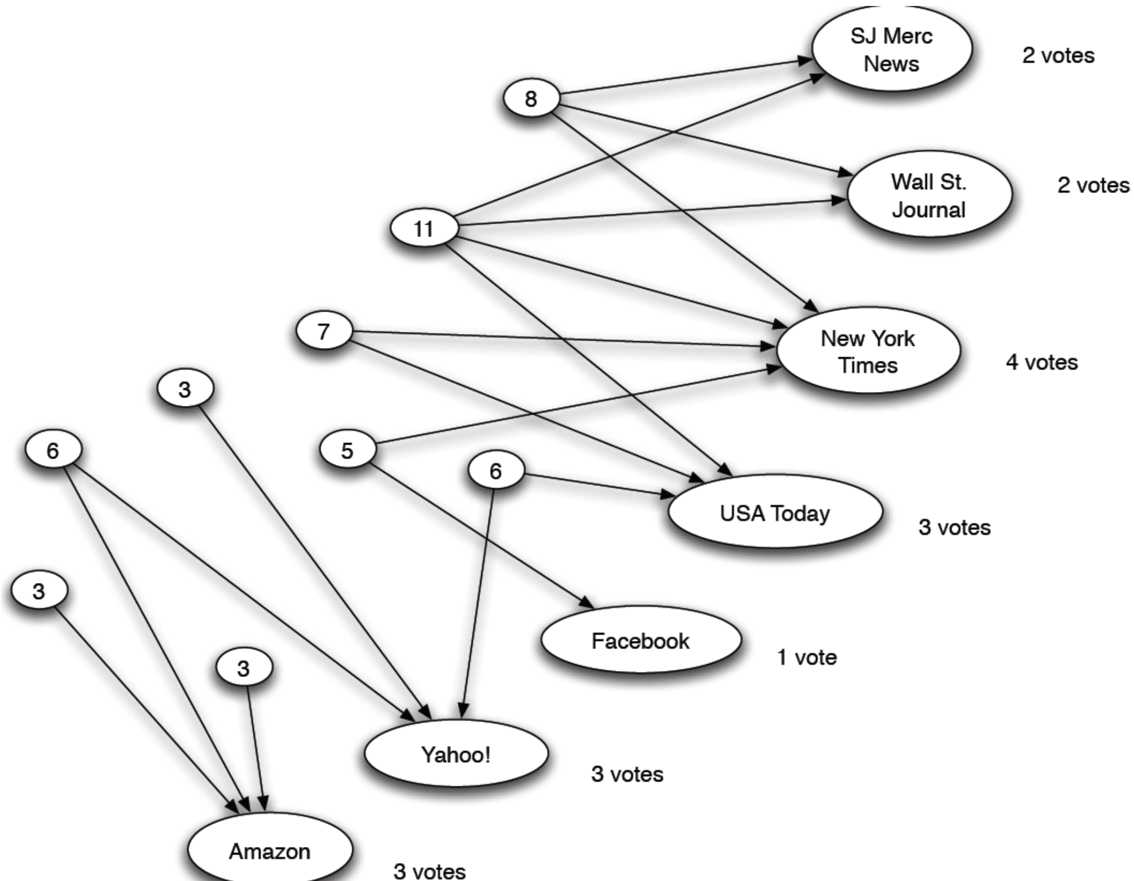
1. Run the **Authority Update Rule**
2. Run the **Hub Update Rule**
3. Repeat ***k*** times the previous steps (2 and 3)

4. At the end, is common to normalize the values of both *auth(p)* and *hub(p)*, due to their tendency to grow and become very large. The normalization consists in dividing the authority score (the hub score) by the sum of the squares of all authority scores (all hub scores).

Link analysis: hubs and authorities
1. HITS algorithm

HITS algorithm: toy example k=1

(Easley and Kleinberg, 2010)



Query *Q*:“newspapers”
k = 3

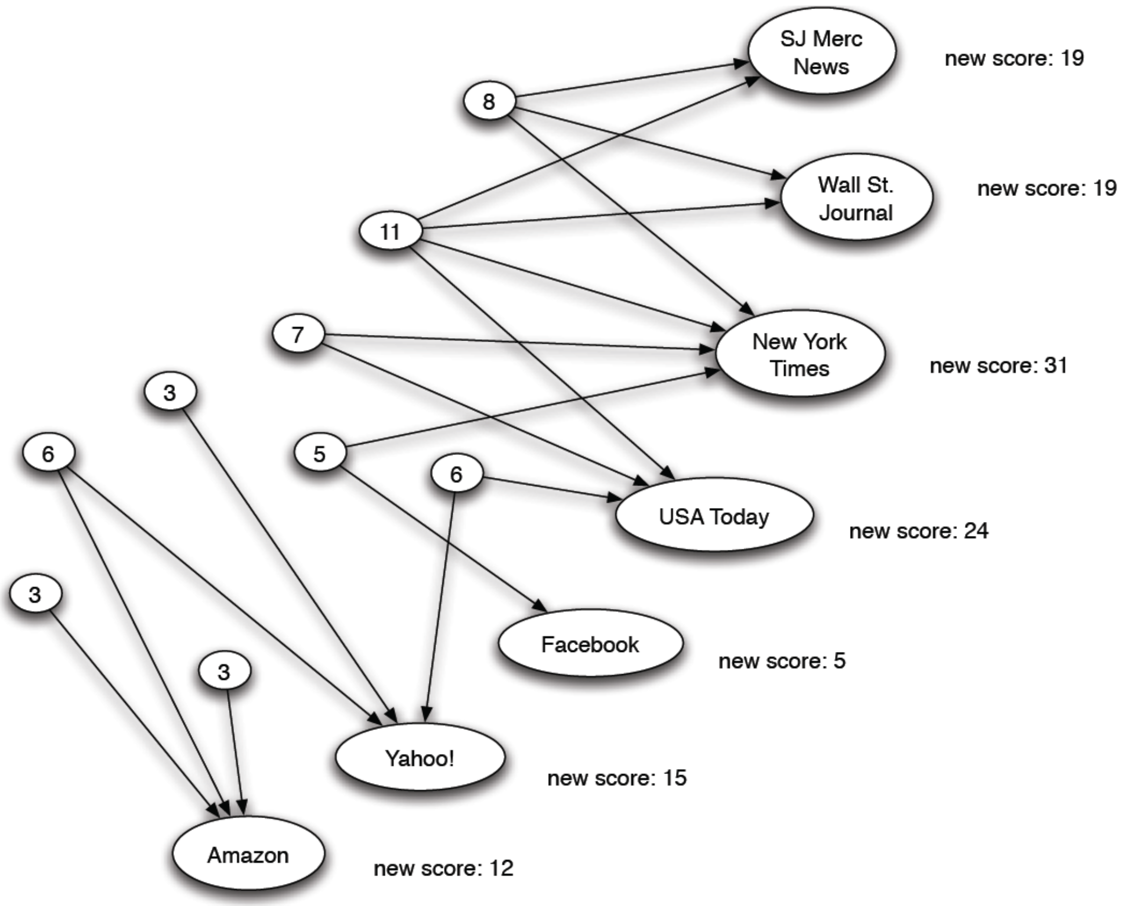
→<u>Hubs</u> are represented by small circles
→<u>Authorities</u> are represented by elongated circles

Each circle has assigned the *hub score* and the *authority score* of each page after the first iteration (k=1) of HITS algorithm

Link analysis: hubs and authorities
1. HITS algorithm

HITS algorithm: toy example k=2

(Easley and Kleinberg, 2010)



Query *Q*:“newspapers”

→Hubs are represented by small circles
→Authorities are represented by elongated circles

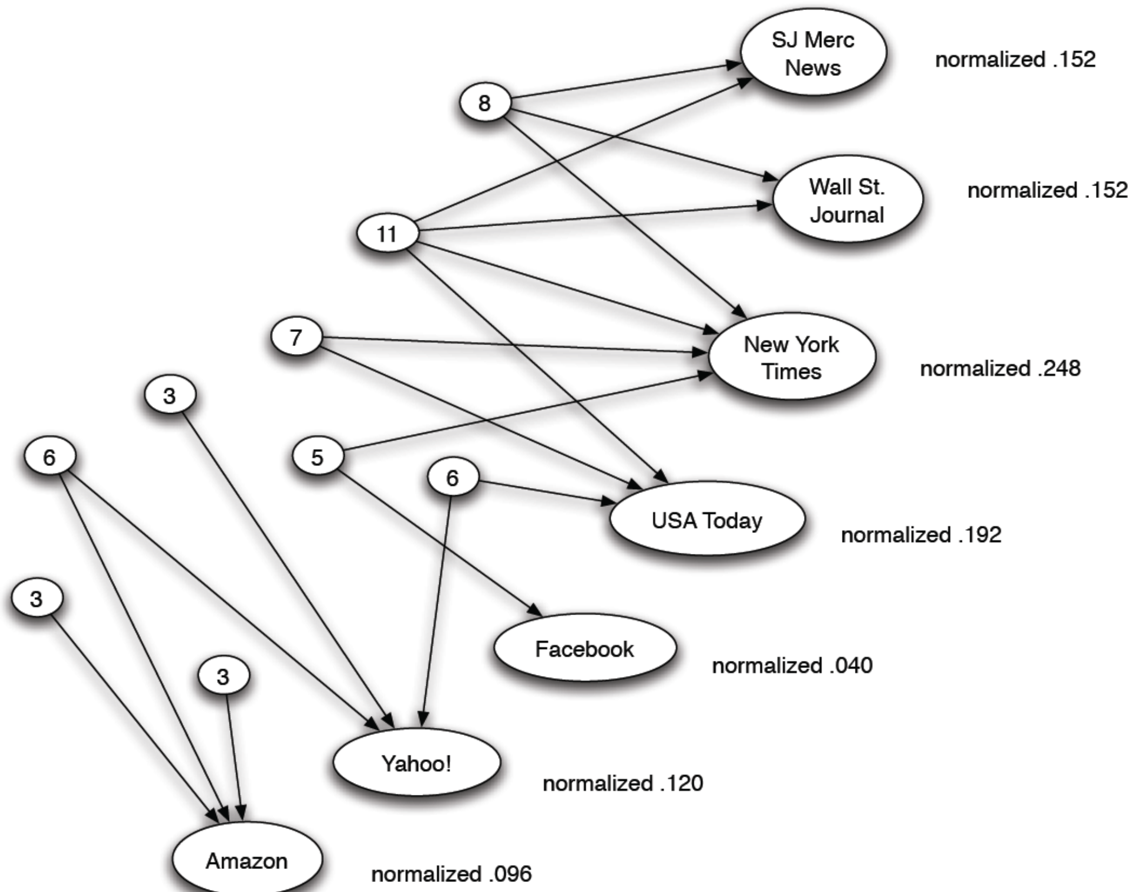
Second iteration of the algorithm (k=2).

E.g. Amazon is cited by 3 hubs, thus the new score of this page will be, according to the ***authority update rule***, the sum of the values of all hubs that point to it (new score=3+3+6=12)

Link analysis: hubs and authorities
1. HITS algorithm

HITS algorithm: toy example k=3

(Easley and Kleinberg, 2010)



Query *Q*:“newspapers”

→<u>Hubs</u> are represented by small circles
→<u>Authorities</u> are represented by elongated circles

Third and last iteration of the algorithm (k=3). Normalization of the returned authority scores.

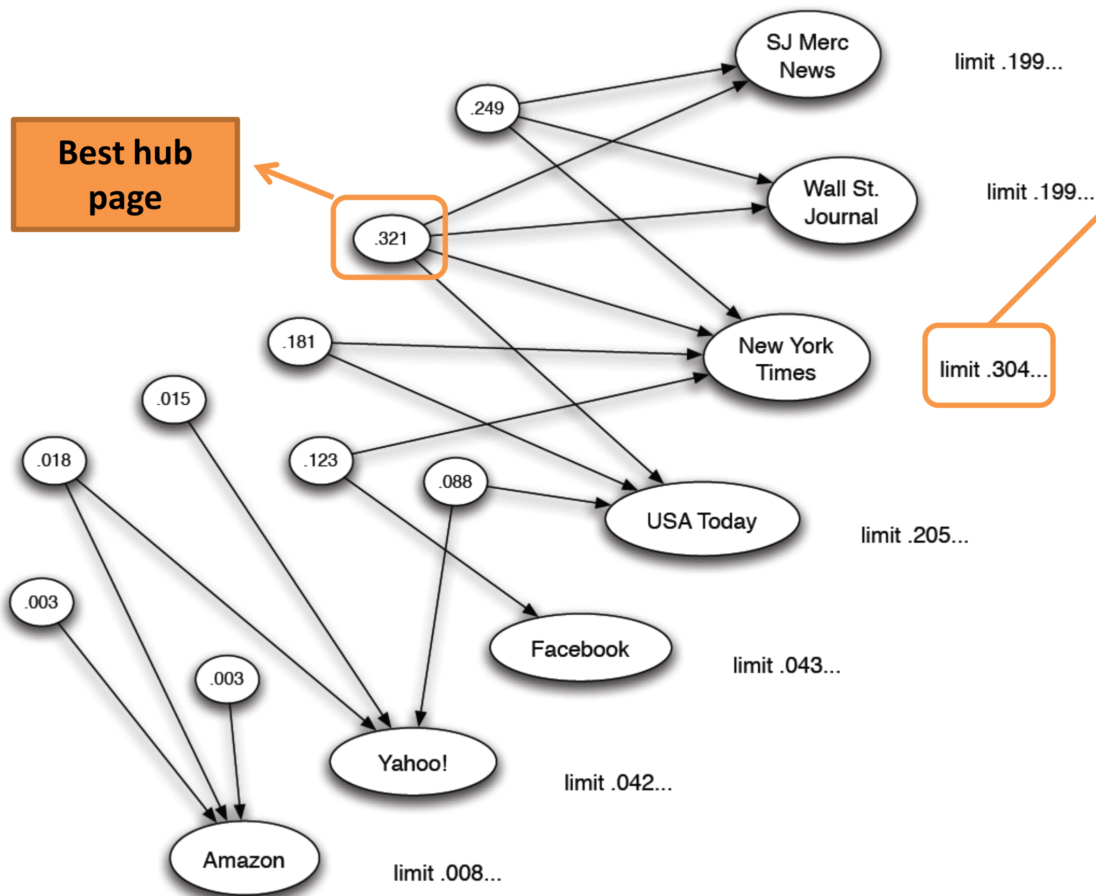
∑*auth(p)*= 19+19+31+24+5+15+12= 125
Norm.auth(Amazon)= 12/125= 0.096
Norm.auth(Yahoo!)=15/125= 0.120
(…)

Link analysis: hubs and authorities
1. HITS algorithm

HITS algorithm: toy example
Convergence of normalized scores

(Easley and Kleinberg, 2010)

Best authority page:
"New York Times"



Query *Q*:"newspapers"

→ If we run HITS algorithm for very large values of *k* (infinite *k*), the normalized scores for hubs and authorities will converge to limits.

→ These limits are represented in the network.

Link analysis: hubs and authorities
2. PageRank algorithm

PageRank algorithm

- **PageRank** (Brin and Page, 1998) is a link analysis algorithm, which is in the basis of Google's search technology, and it is built upon the concept of eigenvector centrality
- The idea of the algorithm is that information on Web can be ranked according to link popularity: the more web pages are linked to a given web page the more popular that web page is
- In this process of weighting web pages, not only the number of links (degree of a node) is important, but also the importance of the web pages linking to them.

Link analysis: hubs and authorities
2. PageRank algorithm

PageRank algorithm

- **Initialization:** in a network of ***n*** nodes, assign a PageRank value of ***1/n*** to each node, and choose the number of iterations ***k***

1. Update the values of each node's PageRank by sequentially applying the following rule:

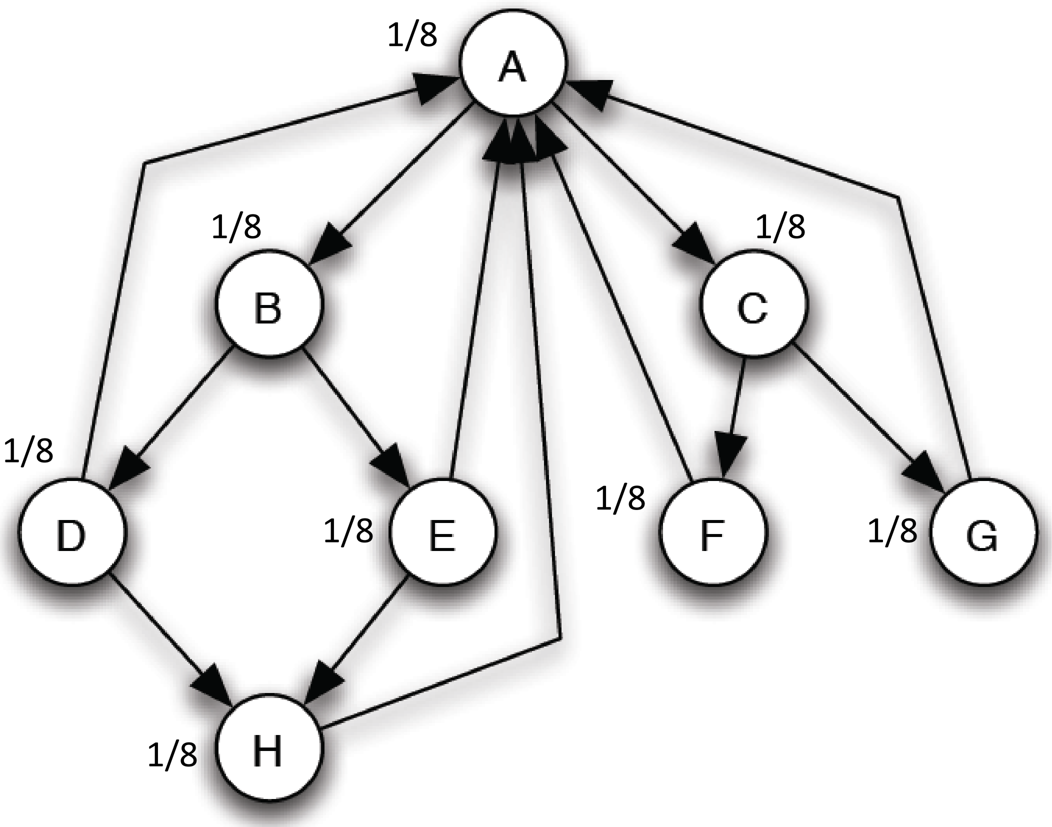
> **Basic PageRank Update Rule:** divide the actual PageRank value of page *p* (or node *v*) by the number of its outgoing links and pass these equal shares to the pages it points to. The update of a node's PageRank value is performed by summing the shares it receives in each iteration.

2. Apply this rule until the ***k-th*** iteration.

Link analysis: hubs and authorities
2. PageRank algorithm

PageRank algorithm: toy example
Initialization

(Easley and Kleinberg, 2010)



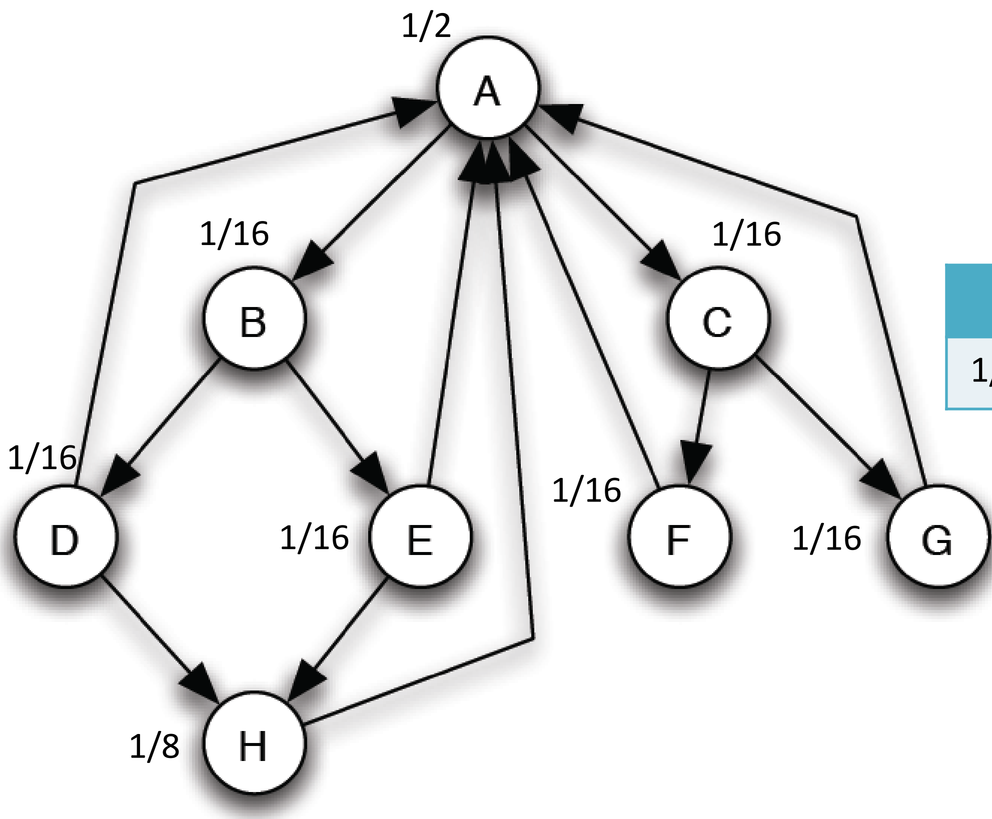
Example: Network comprised of 8 Web pages (n=8)

→ At the <u>initialization step</u>, all pages are assigned a PageRank (PR, from now on) value of 1/8

Link analysis: hubs and authorities
2. PageRank algorithm

PageRank algorithm: toy example
k=1

(Easley and Kleinberg, 2010)



→ At the end of the <u>first iteration of the algorithm</u> we obtain new PR values by applying the *Basic PageRank Update Rule*

→ To apply the rule, first is necessary to compute the **shares** of all nodes

A	B	C	D	E	F	G	H
1/16	1/16	1/16	1/16	1/16	1/8	1/8	1/8

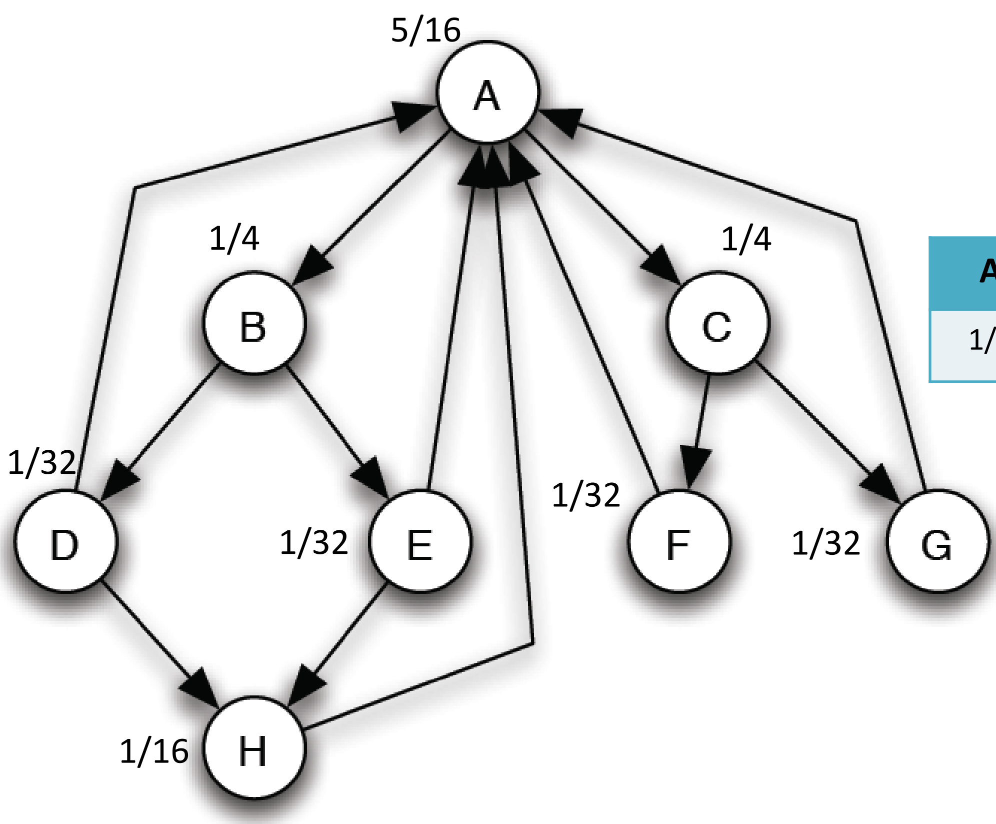
→ Then, for each node we sum all shares the node receives; the result of this sum will be its new PR value

e.g.PR(A)=(1/16)+(1/16)+(1/8)+(1/8)+(1/8)=4/8 =1/2

Link analysis: hubs and authorities
2. PageRank algorithm

PageRank algorithm: toy example k=2

(Easley and Kleinberg, 2010)



→ The rule is applied iteratively until the convergence of PageRank values, or until the *k-th* iteration

→ **Shares of nodes for k=2:**

A	B	C	D	E	F	G	H
1/4	1/32	1/32	1/32	1/32	1/16	1/16	1/8

e.g. PR(A)=(1/32)+(1/32)+(1/8)+(1/16)+(1/16)= 5/16

Outline

Properties of real-world complex networks

Real-world networks are non-random and non-regular graphs with unique features.

❖ Examples of such networks are:

- ✓Social networks
- ✓Information or knowledge networks
- ✓Technological networks
- ✓Biological networks

❖ These unique features can be summed up by the following **properties**:

1. Small-world effect
2. Transitivity or Clustering
3. Power-law degree distributions
4. Network resilience
5. Mixing patterns
6. Community structure

Properties of real-world complex networks

1. Small-world effect

- Stanley Milgram, an American social psychologist, was the first to point out the existence of ***small-world effects*** in real social networks, through a series of famous experiments which are today known as the *Milgram experiment* (1960's)

- To probe the distribution of the path lengths, Milgram asked some random participants (about 300) to pass a letter to someone they knew in a first-name basis in an attempt to get it to an assigned target person.

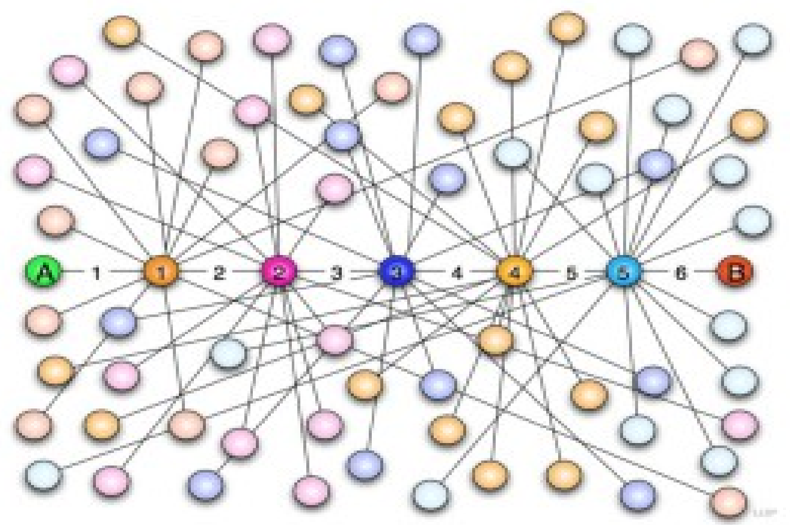
- The goal of the experiments was to test the speculative idea that <u>pair of apparently distant individuals in most networks are connected by a few number of acquaintances</u>

- With these experiments Milgram was able to show that the median path length of the paths that succeeded in reaching the target person was 6

Properties of real-world complex networks

1. Small-world effect

These findings are on the basis of the ***six degrees of separation*** concept, that states that everyone is, on average, six steps away from any other person in the World



- The overall conclusion of Milgram and its colleagues has been accepted in a broad sense, since it is believed that <u>social networks tend to be characterized by very short paths between randomly chosen pairs of people</u>.

- These findings have important consequences, for instance, in the speed at which information and diseases spread.

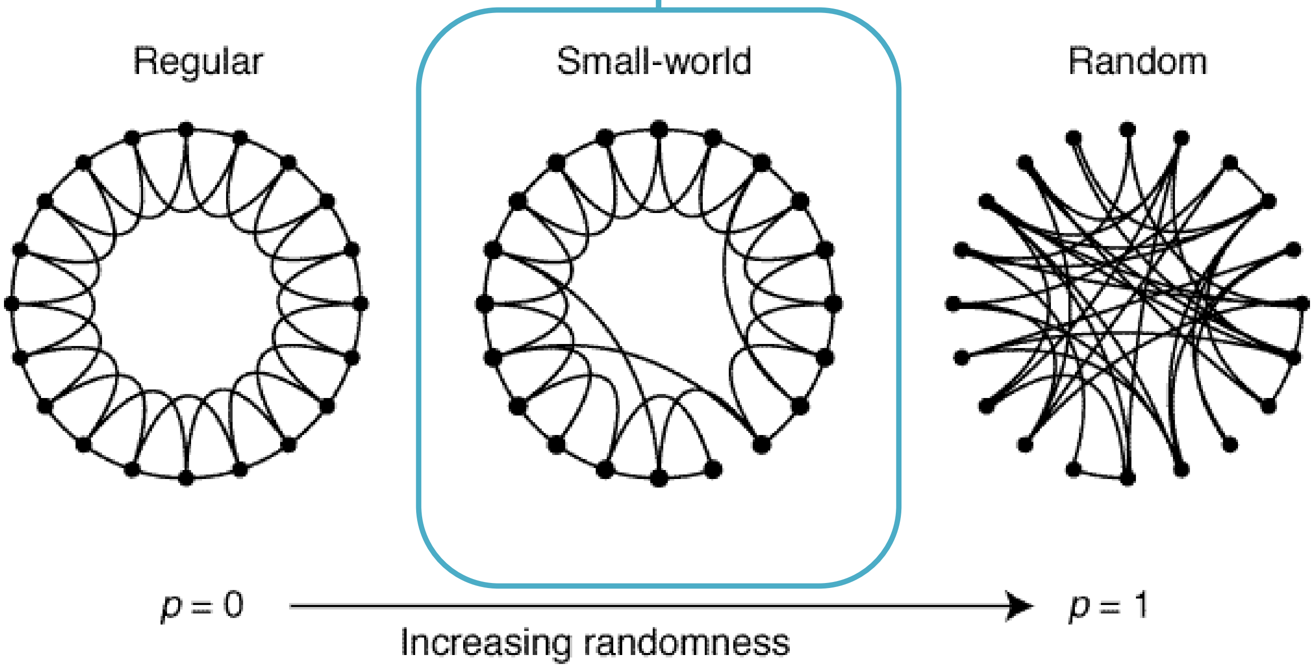
Properties of real-world complex networks

1. Small-world effect

Order coexists with disorder

(model proposed by Watts and Strogatz, in 1998)

Perfect order

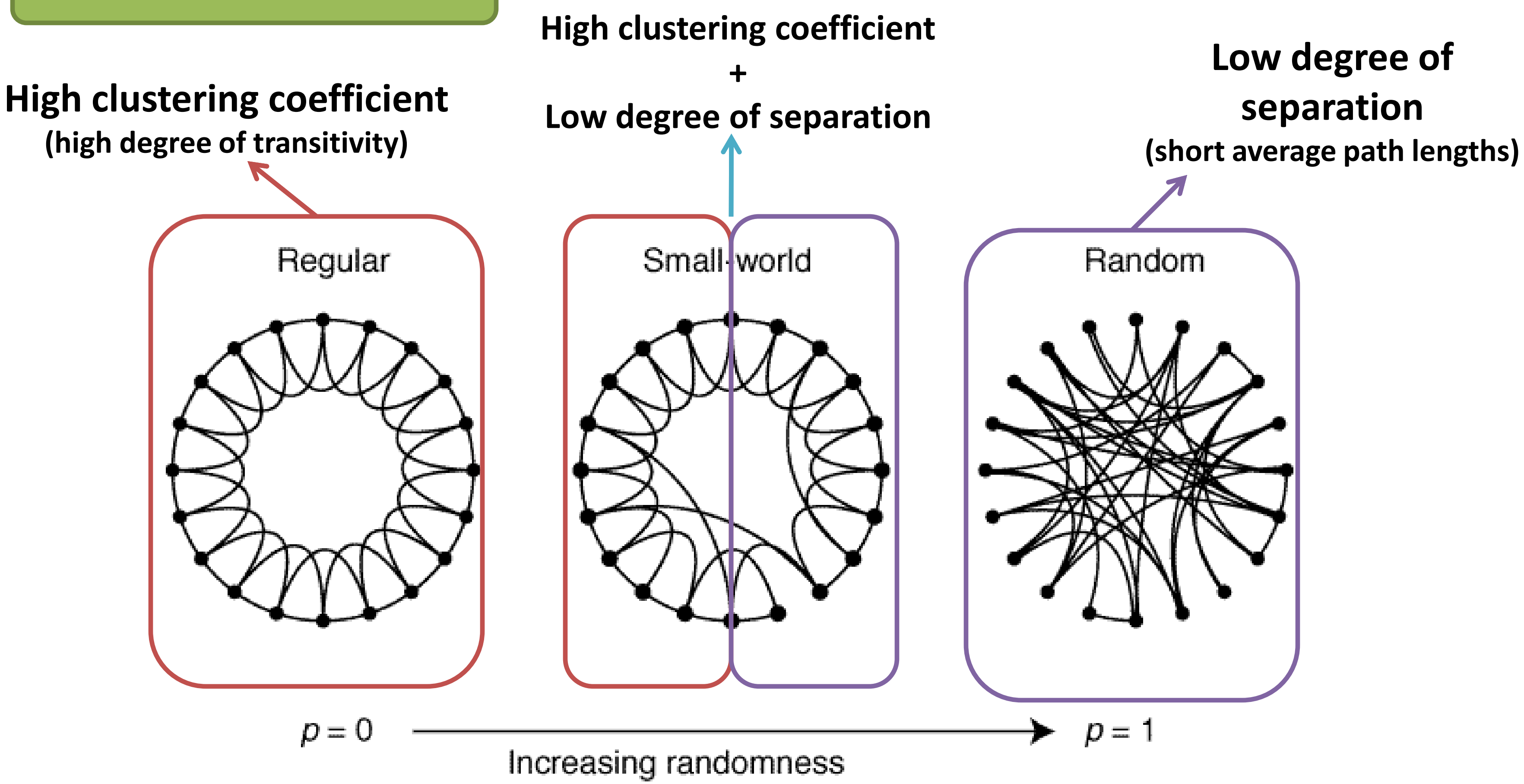


Total Chaos

(model first proposed by Erdös and Rényi, 1959)

Properties of real-world complex networks

1. Small-world effect



Properties of real-world complex networks

2. Transitivity or Clustering

• In real-world networks, especially in social ones, there is a high probability of finding complete sub-networks, within larger networks, where all nodes are connected to each other (everyone knows everyone)

3. Power-law degree distributions

Degree distribution: probability distribution of the degrees of nodes over the whole network or, in other words, a histogram that shows the fraction of nodes in the network that have degree ***k*** ($k=1,2,3,\ldots,k_{max}$)

• Real networks have degree distributions that are quite different from other networks, such as random graphs

Properties of real-world complex networks

3. Power-law degree distributions

Random graphs

<u>Homogeneous</u> degree distribution

Binomial degree distribution (Poisson in the limit of graph size)

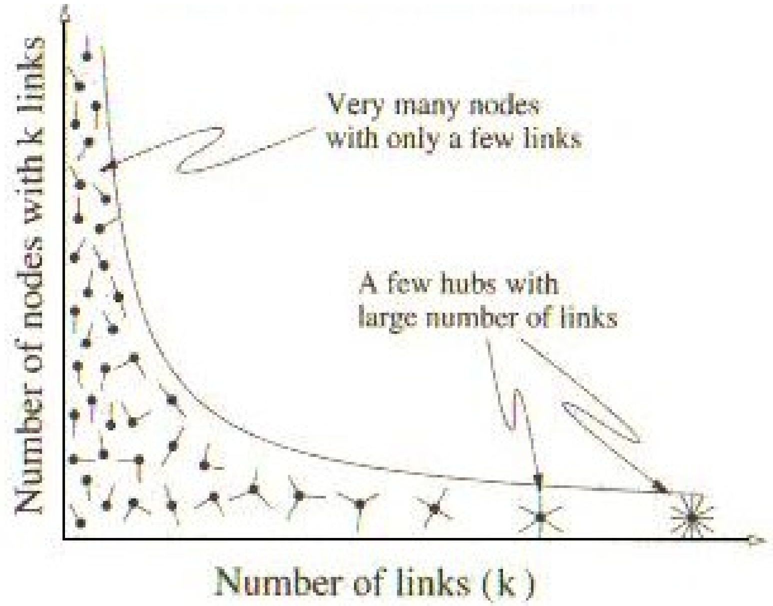
Real-world networks

<u>Heterogeneous</u> degree distribution

Power-law degree distribution (at least asymptotically)

Scale-free networks

Power-law degree distribution



Properties of real-world complex networks

3. Power-law degree distributions

Scale-free networks (Barabási and Albert, 1999)

- **Scale-free networks** are networks whose degree distribution follows a **power-law** (e.g. world wide web, citation networks, biological networks, some social networks, etc.)

- Power-law distributions usually arise when the amount you get of something depends on the amount you already have; in topological terms this reflects in a network with few highly-connected nodes and a large number of nodes with low degree

- The mechanism behind this kind of degree distribution was referred to as being "*the rich-get-richer and the poor-get-poorer*" strategy or, equivalently, the *"Matthew's effect"*; Prince called it "*cumulative advantage*" and, more recently, Barabási and Albert used the expression "*preferential attachment*"

Properties of real-world complex networks

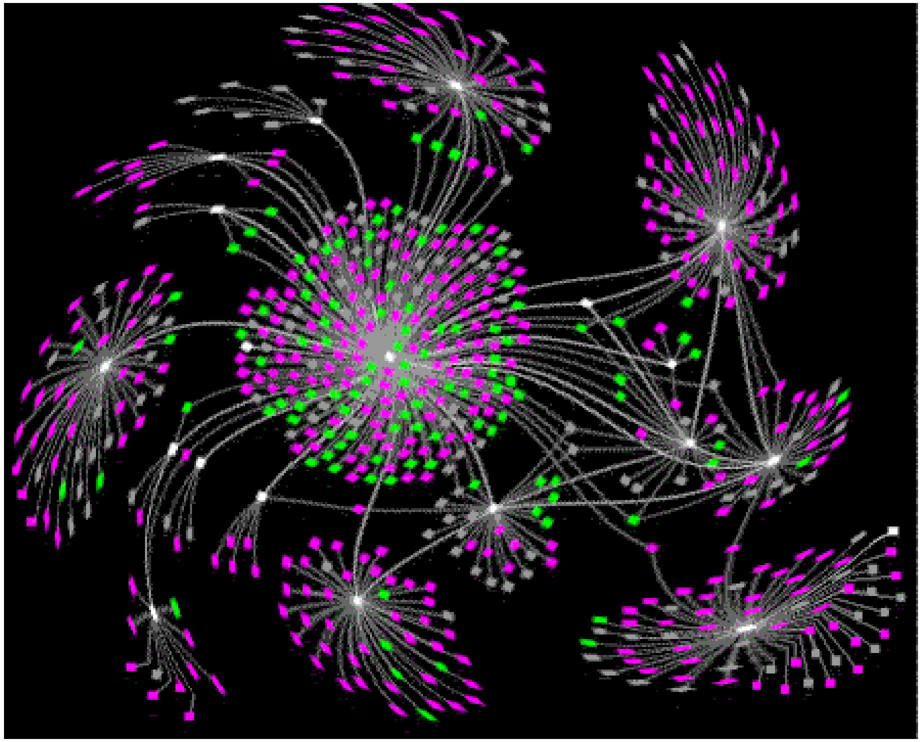
3. Power-law degree distributions

Scale-free networks

(Barabási and Albert, 1999)

- The mechanism of *"preferential attachment"* is easily explained by the fact that new nodes (e.g. individuals) entering the network tend to connect to well-connected nodes, which are often associated to central and prestigious positions (e.g. individuals with more status, popularity, knowledge, money etc.) in the network.

- These highly-connected nodes are known as **hubs**



Properties of real-world complex networks

4. Network Resilience

- Measures the impact on the connectivity of the network when one or more nodes are removed

- Most networks are robust against ***random vertex removal*** but considerably less robust to ***targeted removal of the highest-degree vertices***

- Also, when ***gatekeepers*** are deleted there are strong changes in the network with respect to the ability of communication between pairs of nodes, since some of them become disconnected.

- In real-world networks, the removal of one single node is not cause for alarm, since it rarely has impact in the original network structure; in such cases, it is more appropriate to test the resilience of a network by removing a certain ***percentage of nodes***.

Properties of real-world complex networks

5. Mixing Patterns

- In some networks, where different types of nodes coexist, is common to observe a certain selectivity in the establishment of connections
- This ***selective linking*** is usually called ***assortative mixing*** or ***homophily*** and a classic example is mixing by race
- Real networks show higher tendencies for assortative mixing

6. Community structure

- Most real social networks show **community structure**
- This property usually arises as a consequence of both global and local heterogeneity of edges' distribution in a graph.
- Thus, we often find high concentrations of edges within certain regions of the graph, that we call ***communities***, and low concentration of edges between those regions

Outline

1. Background
2. Practical applications
3. Graph Theory:
    1. Types and representation of graphs
    2. Cliques
4. Fundamental concepts of SNA
5. Statistical measures to analyze networks
6. Link Analysis: hubs and authorities
    1. HITS algorithm
    2. PageRank algorithm
7. Properties of real-world networks
8. **Community detection**

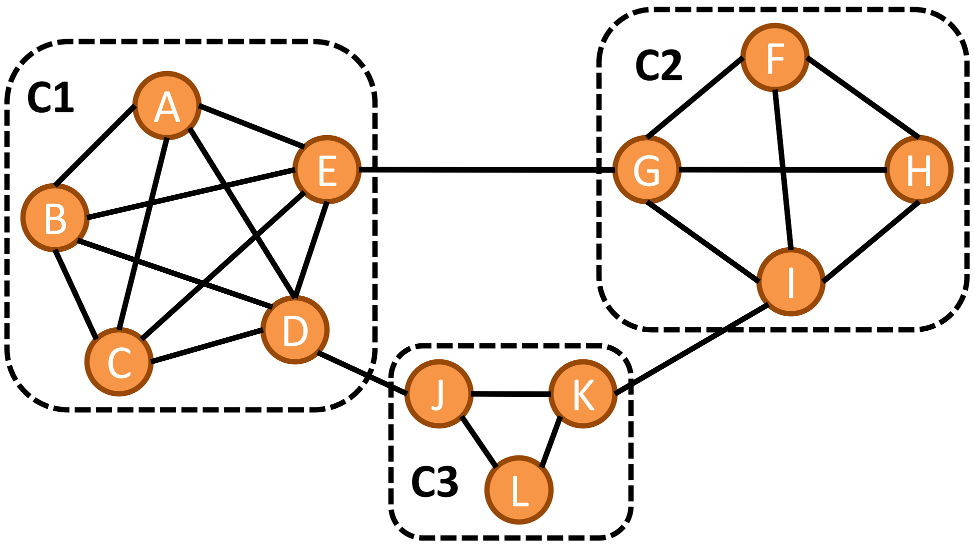
Community detection

Communities, Modules or Clusters

What are network communities?

- Similar groups of nodes
- Densely connected groups of vertices in the network, with sparser connections between them

Examples: families, work groups, circles of friends, topic-related web pages, customers of a given product, etc.



The graph on the left has 3 communities:

C1={A,B,C,D,E}
C2={F,G,H,I}
C3={J,K,L}

Community detection

Two main lines of research

Graph Partitioning

Origins:	Computer Science
Motivation:	Improve the computation in parallel computing environments by minimizing the communication between processors
Example:	Minimum cut algorithm

Community structure detection
Blockmodeling
Hierarchical Clustering

Origins:	Sociology
Motivation:	Simplify the analysis of social phenomena through the arrangement of people according to their similarities
Example:	Girvan-Newman algorithm

Community detection

❖ **Several methods and algorithms were proposed to address the problem of finding communities in networked data:**

- ✓ Minimum cut
- ✓ Hierarchical clustering using *cosine similarity*, *Jaccard index*, *Euclidean distance*, *Hamming distance* as measures of similarity
- ✓ Girvan-Newman algorithm
- ✓ Walktrap
- ✓ Modularity Optimization using *greedy techniques*, *simulated annealing,* among others
- ✓ Clique percolation method for finding overlapping communities
- ✓ Blockmodeling
- ✓ …

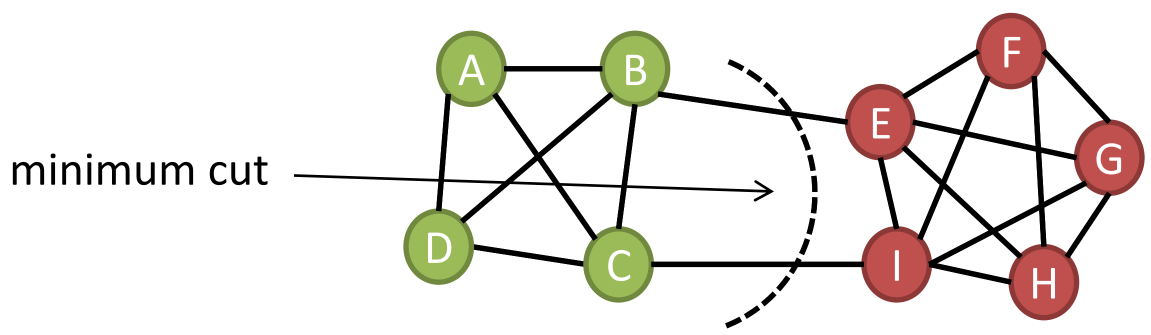
Community detection: Minimum-Cut Algorithm

Minimum-Cut **(Ahuja et al., 1993)**

- **Idea:** divide the network into two communities by minimizing the number of edges (or the sum of the weights, in weighted networks) running through unlike groups, also known as the *cut-size*

- **For k>2**: implement the strategy of iterative bisecting (in the second and further iterations, groups are sequentially divided into two sub-groups)

- **Drawback:** produces groups of unbalanced sizes



Community detection: Girvan-Newman algorithm

Girvan-Newman algorithm **(Girvan and Newman, 2002)**

- Divisive hierarchical algorithm, based on a *top-down* approach, since it deconstructs the initial full graph into progressively smaller connected pieces, until there are no edges to remove and each node represents itself a community

- **Idea:** identify edges that connect vertices belonging to different communities (the so-called ***bridges***) and, iteratively, remove them from the graph

- **Criterion:** *edge betweenness* is the adopted measure to identify and delete *bridges* (edges with high betweenness are removed), since it is able to identify edges that lie in a large number of shortest paths between vertices

- **Drawback:** high computational cost, being only suitable for networks of moderate size

Community detection: Girvan-Newman algorithm

Girvan-Newman algorithm **(Girvan and Newman, 2002)**

Algorithm

Input: full graph
Output: hierarchical structure that can be represented by means of a dendrogram

1. Compute the betweenness of all edges in the network
2. Remove the edge with highest betweenness
3. Repeat the previous steps until there are no edges to remove in the graph

To see how this algorithm works visit http://igraph.sourceforge.net/screenshots2.html

Selecting the number of communities

❖ This algorithm returns a set of possible solutions, however it does not indicates which one is the best.

❖ To select the best partition is common to compute the **modularity** of each network's division and select the one with higher modularity.

Community detection: Evaluating Community Quality

Modularity **(Girvan and Newman, 2004)**

- **What is modularity?** Modularity ***Q*** is a quality function that quantifies the quality of a given division of the network into communities.

- **Basic Idea:** a network has meaningful community structure if the *number of edges* between communities is **fewer than expected** on the basis of random choice.

- **Modularity values:** Modularity can be either *positive* or *negative, Q ϵ* [-1,1].
  → If *positive*, then there is possibility of finding community structure on the network
  → If the values are not only *positive*, but also *large*, then the corresponding partition may reflect the real community structure of the network. To assure meaningful communities modularity should be equal or higher than ***Q>=0.3*** (Clauset et al.,2004).

$$Q = \frac{1}{2m}\sum_{ij}\left[A_{ij} - \frac{k_i k_j}{2m}\right]\delta(c_i, c_j),$$

$\frac{k_i k_j}{2m}$ - expected number of edges falling between vertices *i* and *j*

$\delta(c_i, c_j)$ - Kronecker delta

m – number of edges

k_i – degree of vertex *i*

c_i – group to which vertex *i* belongs

A_{ij} – entry of the adjacency matrix that gives the number of edges between vertices *I* and *j*

Community detection: Louvain algorithm

Louvain method **(Blondel et al., 2008)**

- Greedy optimization method, that performs an agglomerative hierarchical modularity optimization

- **Process:** the algorithm comprises two phases
  - ✓ First, it looks for local optima by minimizing modularity in a local way
  - ✓ Then, it aggregates nodes belonging to the same community and creates a new network where each node represents one of the previously found communities

- **Advantages:** achieves good performance in large networks with low computational cost

- **Drawback:** is order-sensitive

Community detection: Louvain algorithm

Louvain method **(Blondel et al., 2008)**

Algorithm

1. **First phase:** consider each node as a single community
   1.1. Compute modularity **Q**
   1.2. Move the isolated node from its community to a neighboring community
   1.3. Compute the gain/loss in modularity yielded by the assignment of the node to this new community
   1.4. If the modularity increases, i.e. there is a gain, keep the node in the "new" community
   1.5. Repeat the process until no further improvements are possible
   Output: the ***k***-th level partition (***k*** is the number of the iteration)

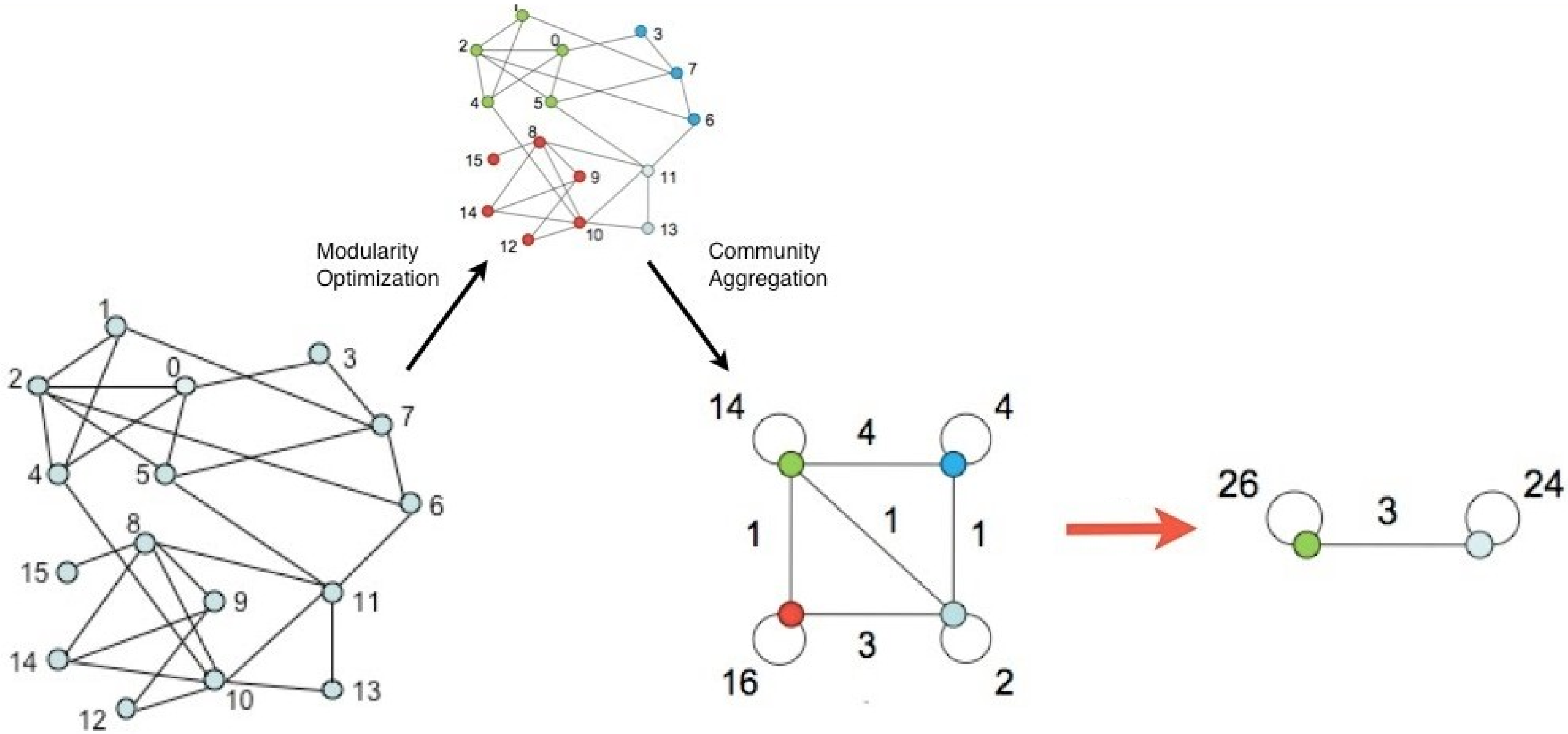
2. **Second phase:** create a *new network* (or *supergraph*), derived from the original one, where each *new node* (or *supervertex*) is the *aggregation of the nodes* assigned to a given community (found in the first phase); two *supervertices* are connected by an edge if there is at least one edge linking two vertices inside the corresponding community.

3. **Repeat steps 1 and 2 until a maximum of modularity is attained**

Community detection: Louvain algorithm

Louvain method **(Blondel et al., 2008)**

Illustration of the process behind the algorithm



Community detection: Challenges

❖ Despite the high number of community detection algorithms, finding communities is still considered a **challenging problem**, due to **two main reasons**:

✓ Typically, the *number of communities* is unknow and has to be determined

✓ Communities are usually *heterogeneous* with respect to size and density.

Some references

- Giorgos Cheliotis, 2010: http://www.slideshare.net/gcheliotis/social-network-analysis-3273045 (accessed in 23th February 2011)
- Wasserman, S. and Faust, K. (1994). *Social Network Analysis: Methods and Applications*. Cambridge University Press, Cambridge.
- Easley, D. and Kleinberg, J. (2010). *Networks, Crowds and Markets: Reasoning about a Highly Connected World*. Cambridge of University Press, New York,USA.
- Rapoport, A. (1953). *Spread of information through a population with sociostructural bias I: Assumption of transitivity*. Bulletin of Mathematical Biophysics, 15(4):523533.
- Granovetter, M. (1973). *The strength of weak ties*. American Journal of Sociology, 78(6):13601380.
- Kleinberg, J. (1999). *Authorative sources in a hyperlinked environment*. Journal of the ACM, 46(5):604632.
- Brin, S. and Page, L. (1998). *The anatomy of a large-scale hypertextual web search engine*. Computer Networks and ISDN Systems, 30(1-7):107117. Proceedings of the Seventh International World Wide Web Conference.
- Watts, D. J. and Strogatz, S. H. (1998). *Collective dynamics of small-world networks*. Nature, 393:440442.

Some references

- Erdös, P. and Rényi, A. (1961). *On the evolution of random graphs*. Bull. Inst. Internat. Statist., 38(4):343347.
- Barabási, A.-L. and Albert, R. (1999). *Emergence of scaling in random networks.*Science,286(5439):50951.
- Ahuja, R. K., Magnanti, T. L., and Orlin, J. B. (1993). *Networks Flows: Theory, Algorithms, and Applications.* Prentice Hall, New Jersey, USA.
- Girvan, M. and Newman, M. E. J. (2002). *Community structure in social and biological networks.* Proceedings of the National Academy of Sciences of the United States of America, 99(12):78217826.
- Blondel, V., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). *Fast unfolding of communities in large networks*. Journal of Statistical Mechanics: Theory and Experiment, 2008(10):P10008.
- Newman, M. E. J. and Girvan, M. (2004). *Finding and evaluating community structure in networks.* Physical Review E, 69(2):026113.
- Clauset, A., Newman, M. E. J., and Moore, C. (2004). *Finding community structure in very large networks*. Physical Review E, 70(6):066111.

Part II

Outline

PART II

1. **Software for Social Network Analysis**
2. Getting started with ***Gephi***: a practical exercise
    1. Extract your Facebook ego-network using netvizz application
    2. Import data to ***Gephi***
    3. Visualize, manipulate and analyze your own network
    4. Find communities and interpret them using your domain knowledge
3. Presentation of the *graph streaming* feature of ***Gephi***

Software for Social Network Analysis

- There is a considerable collection of software and packages for SNA

- **Each software has one, or more, specific functionalities, such as:**
    - → Creation of networks
    - → Visualization and manipulation of networks
    - → Qualitative and quantitative/statistical analysis of networks
    - → Community detection
    - → Predictive analysis (peer influence/contagion modeling, homophily models, link prediction)

(...)

Softwares for social network analysis

Pajek	Software for the analysis and visualization of large scale networks
Gephi	Interactive visualization, manipulation and exploration platform for all kinds of networks; ideal platform for *dynamic network analysis*
Ucinet	Social network analysis tool
CFinder	Software for finding and visualizing overlapping communities in networks; has implemented the Clique Percolation Method
Tulip	Information visualization framework dedicated to the analysis and visualization of relational data; appropriate for large scale networks (can manage 1 million of nodes and 4 million of edges)
NetMiner	Commercial software for networks analysis and visualization; analysis of large networks
R	There are several packages available for network analysis and SNA: `tnet`, `statnet`, `sna`, `igraph`, etc.

Software for Social Network Analysis

Gephi.org

Like Photoshop™ for graphs

"The goal is to help data analysts to make hypothesis, intuitively discover patterns, isolate structure singularities or faults during data sourcing. It is a complementary tool to traditional statistics, as visual thinking with interactive interfaces is now recognized to facilitate reasoning. This is a software for Exploratory Data Analysis."

Limitations:

- Not prepared to deal with large scale networks (Gephi can only manage approximately 150.000 nodes) → use **Tulip™** instead
- Community detection module is still at an experimental stage
- Does not allow the export of 3D networks

Outline

PART II

1. Software for Social Network Analysis
2. **Getting started with *Gephi*: a practical exercise**
    1. **Extract your Facebook ego-network using netvizz application**
    2. **Import data to *Gephi***
    3. **Visualize, manipulate and analyze your own network**
    4. **Find communities and interpret them using your domain knowledge**
3. Presentation of the *graph streaming* feature of ***Gephi***

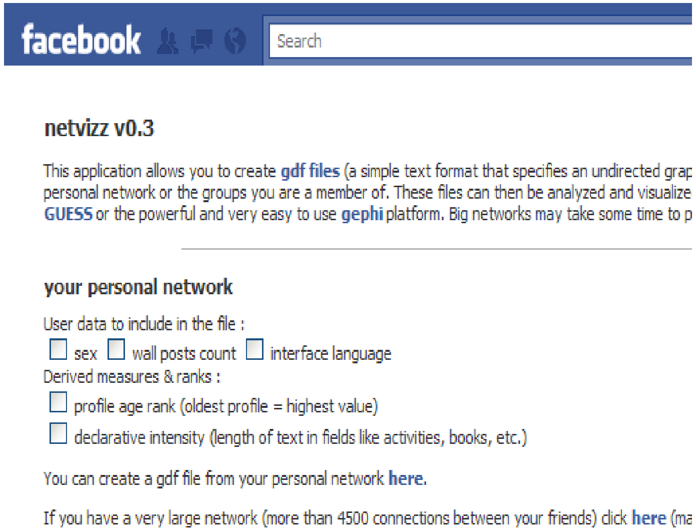
Getting started with Gephi: a practical exercise

1. Extract your Facebook ego-network using **netvizz** application

- Create, visualize and analyze your own Facebook™ friendship social network
- Note that you will extract a network that is organized according to your point of view: it is an *ego-network* (without ego, since you don't appear as a node)

Steps to extract your Facebook ego-network:

1. Login into your Facebook™ account
2. Go to http://apps.facebook.com/netvizz/ or simply look for the **netvizz** application using the *search bar*
3. Select the additional information you want to include in your data (e.g., gender, wall posts..) and click in **here** hyperlink
4. Save data by right clicking in **gdf file** hyperlink that appears after a few seconds
5. Open Gephi™ and import data as an **undirected graph**



Getting started with Gephi: a practical exercise

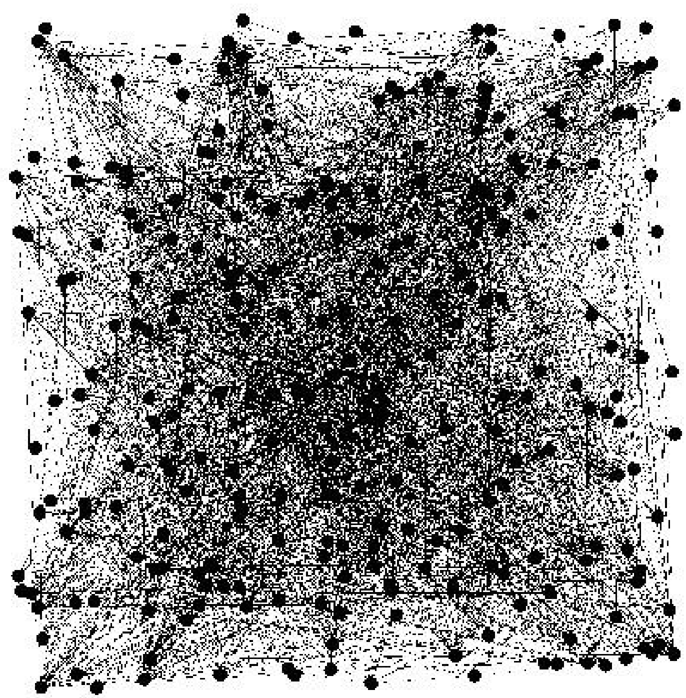
2. Import data to Gephi

- To import your network data to Gephi go to **File** ->**Open** and select the .gdf file you saved from *netvizz* application

- When the **Import Report** window appears, select Graph type: undirected; here you can also see the number of nodes **(order)** and the number of edges **(size)** of your network

- The imported network will look like a large tangle of lines

- We can improve the aspect of the network by changing its layout in the **Layout module**



Node - Facebook friends
Edges – there is a connection if two of your friends are also Facebook friends

Getting started with Gephi: a practical exercise

3. Visualize, manipulate and analyze your own network

- From the Layout module choose **Force Atlas** on the middle left side of the window

- Increase the value of the parameter **Repulsion Strength** to 5000 and select the **Adjust by sizes** box

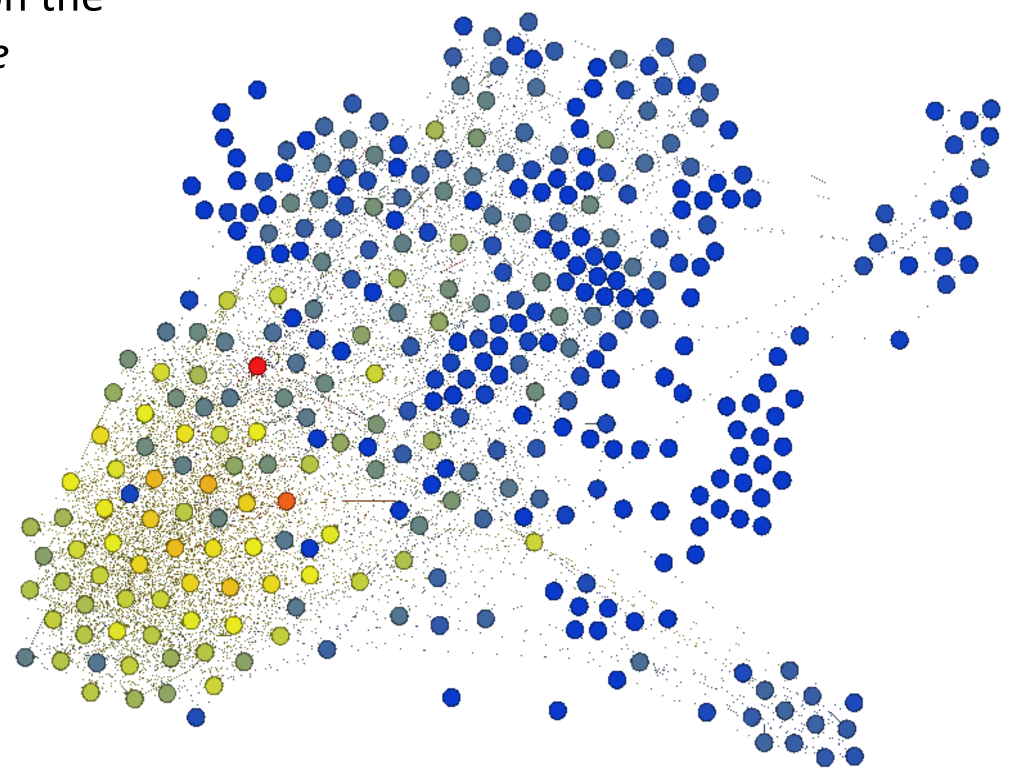
- Click **Run**, wait a little and **Stop** the process when the network's appearance becomes more understandable

- Try also the **Fruchterman Reingold** layout and compare (to continue the exercise choose the layout that you consider most appealing)

Getting started with Gephi: a practical exercise

3.1. Visualize node **degree**

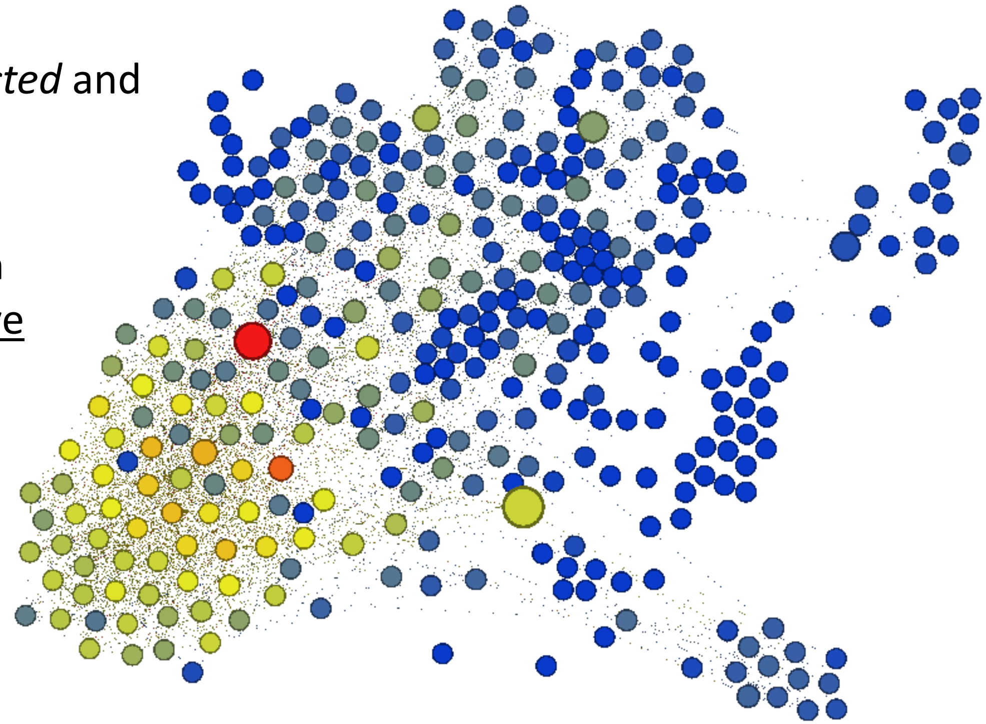
- On the right-side of the window, click on the **Statistics** tab, and run the *Average Degree*
- From the **Ranking** module, located on the top-left side of the drop-down menu choose **Degree**
- Slide the mouse over the **gradient bar** and, for each triangle, select a different color for each side of the *range* (e.g. blue for lower degrees, yellow for intermediate degree and red for higher degrees)
- Select **Apply** to see the result (now high degree nodes are red colored and small degree nodes are blue colored)



Getting started with Gephi: a practical exercise

3.2. Visualize: node **betweenness**

- Go back to the **Statistics** tab and run the *Network Diameter* option
- In the window that pops up select *Undirected* and click OK
- A **Graph Distance Report** will appear with some statistical measures computed for each node of the network; if you want you can save these reports for further analysis
- **Close** the report window and go back to the top-left **Ranking** module, but this time choose the diamond icon (top-right) and the **Betweenness** measure in order to adjust the nodes' size according to their betweenness score



- Set **Min size** and **Max size** to, for instance, 60 and 120, and click **Apply**

Getting started with Gephi: a practical exercise

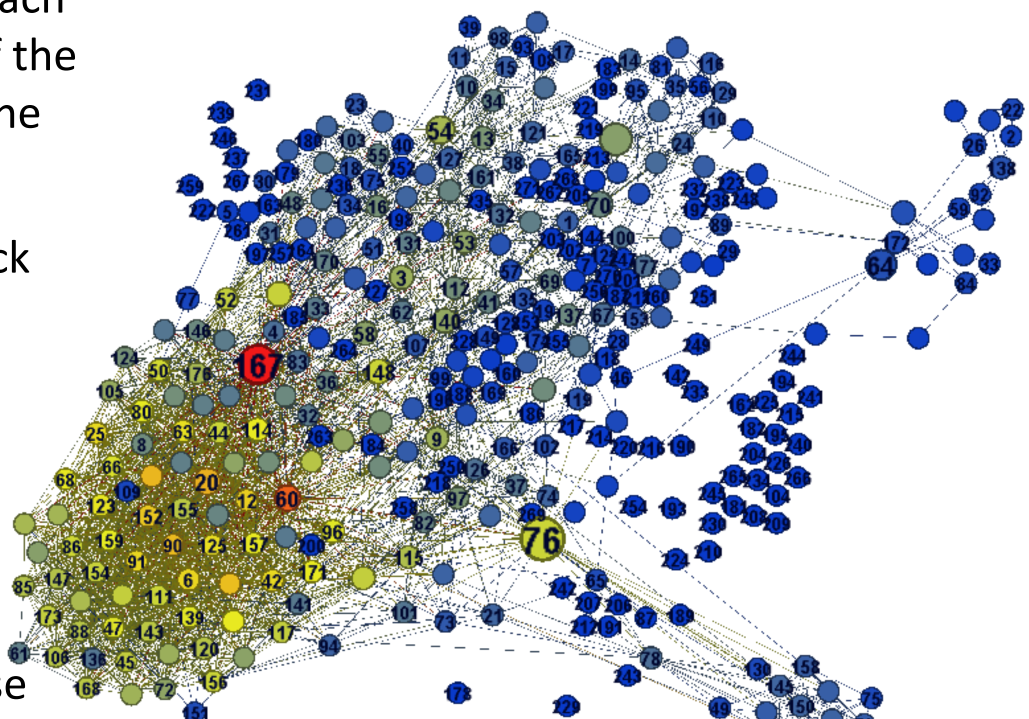
3.3. Visualize: node **labels** and edge **thickness**

- To identify the nodes, i.e. to know to each Facebook friend corresponds each one of the nodes, press the bold black **T** located at the toolbar on the bottom of the window

- In the same toolbar, also press the black **A** to adjust the label size to **Node Size**

- It is possible to change the font of the labels by clicking upon the ***Arial Bold,20*** and also its color by clicking on the **black square** of the right-side of the toolbar

- To adjust the **thickness** of the edges use the left **slider** of the toolbar and move it to the right direction in order to increase it



3.4. Exercise: explore the **Statistics** tab

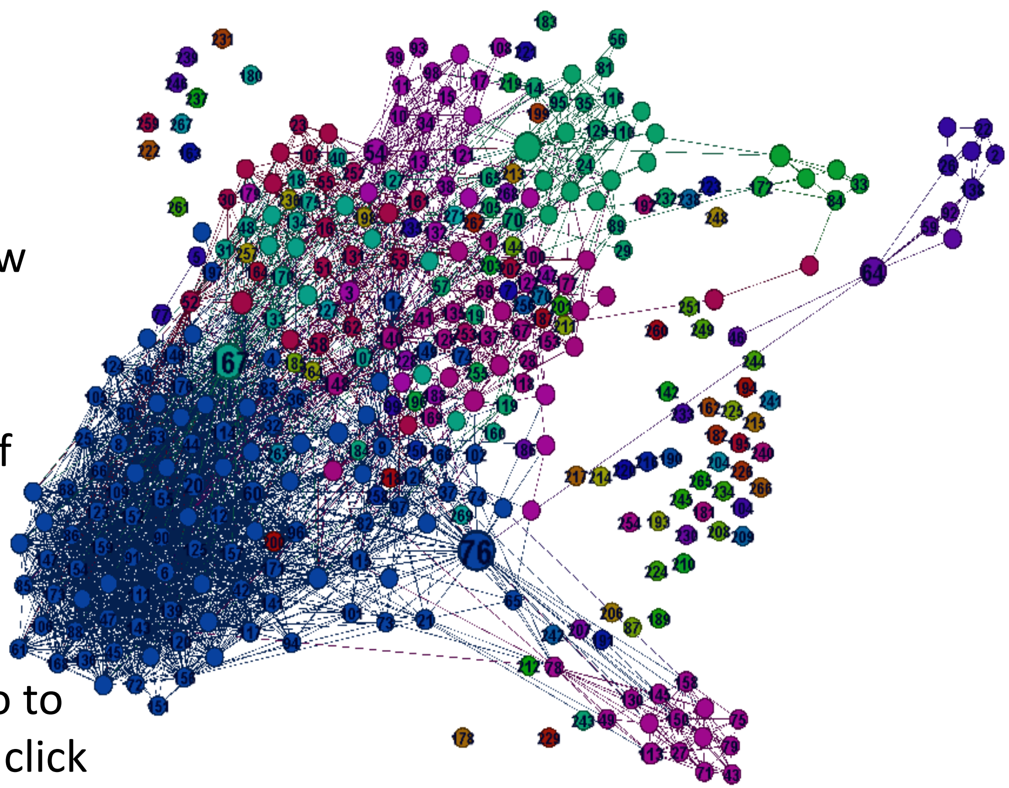
Explore the *Statistics* tab and extract some basic knowledge of your ego-network:

- Radius and Diameter
- Number of shortest paths
- Average path length
- Density of the network: compute the density by hand and compare the obtained value with the density returned by Gephi
- Average degree of the network
- Friends with highest and lowest degree. Do you obtain the same results with eigenvector centrality measure?
- Do you identify any bridges/gatekeepers in the network?
- Number of weakly connected components
- Does your network show the properties of a *small-world* network? Justify.
- Is your network *scale-free*? Justify.
- Export the statistical measures of your network to an Excel file

Getting started with Gephi: a practical exercise

4. **Find communities** and interpret them using your domain knowledge

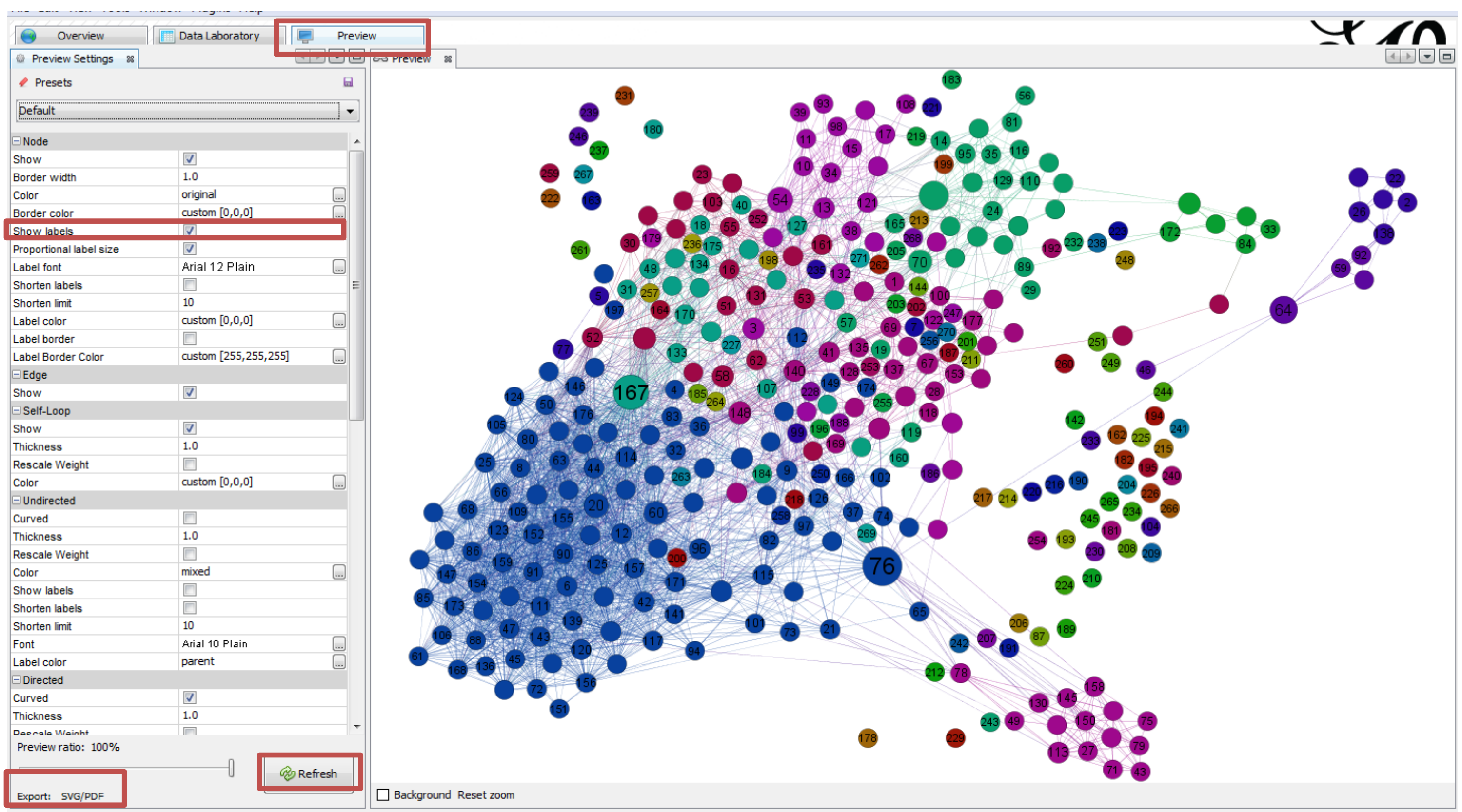
- To discover communities of friends in your network go back to the **Statistics** tab on the right and **Run** the **Modularity** option
- Choose *randomize* on the popup window and click OK
- The **Modularity Report** that appears gives you information about the number of found communities and the modularity of this partition (Gephi uses the **Louvain Method** to detect communities)
- To visualize the detected communities go to the **Partition** module on the top left menu, click on the **Refresh** arrows and select **Modularity class** from the list; then, click **Apply**



Getting started with Gephi: a practical exercise

4.1. **Export** your Facebook network to a PDF file

Go to the **Preview** tab **->** select the **Show Labels** box (<u>Node</u> section) **->** Click **Preview** on the bottom left **->** Choose to **Export** (in .pdf or .svg) on the left of Preview button



Getting started with Gephi: a practical exercise

4.2. Exercise: interpretation of the detected communities

- Do your communities make sense? Do they reflect different social groups you belong to?

- Analyze the number of isolate communities. What does this mean in the context of your network?

- Analyze the relationship between the size of the nodes and the found communities. Do you find any **gatekeeper/broker**?

Outline

PART II

1. Softwares for Social Network Analysis
2. Getting started with *Gephi*: a practical exercise
    1. Extract your Facebook ego-network using netvizz application
    2. Import data to *Gephi*
    3. Visualize, manipulate and analyze your own network
    4. Find communities and interpret them using your domain knowledge
3. **Presentation of the *graph streaming* feature of *Gephi***

Graph Streaming feature of Gephi

- Gephi™ version 0.7 has available a *graph streaming* plugin that allows import and visualization of streaming graph objects in real-time

- *Graph streaming* plugin is based on the idea that graphs are not static objects and may change continuously

- To explore it you just need to open Gephi, connect to a master and start receiving graph data in real-time

- This plugin is not yet complete and improvements will be released in the next version - Gephi™ version 0.8

Demonstration of the *graph streaming* plugin, using Amazon.com library data:

http://gephi.org/2010/gsoc-2010-mid-term-graph-streaming-api/