

From social theories to models

Tutor: Lisette Espín-Noboa

Overview

Time: 10:30 - 12:30

10:30 - 11:00

Social theories

- Popularity
- Similarity
- Friend-of-friend
- Activity

Network properties & structure

11:00 - 11:30

Network models: Undirected

- Barabasi-Albert Preferential Attachment (PA)
- PA with homophily (PAH)
- PA with triadic closure (PATC)
- PAH with triadic closure (PATCH)

11:30 - 12:00

Network models: Directed

- Directed Homophily (DH)
- Directed Preferential Attachment (DPA)
- DPA with homophily (DPAH)

12:00 - 12:30

Exercise #1

Social theories

Literature

Non-exhaustive list of material covered in this section.

1. Jackson, M. O. (2008). Social and economic networks (Vol. 3). Princeton: Princeton university press.
2. Albert-László Barabás (2016). Network Science. (available online as an interactive book)
3. Jackson, M. O. (2019). The human network: How your social position determines your power, beliefs, and behaviors. Vintage.
4. Stadfeld, C., & Amati, V. (2021). Network mechanisms and network models. In Research Handbook on Analytical Sociology (pp. 432-452). Edward Elgar Publishing.
5. Gamper, M. (2022). Social Network Theories: An Overview. Social Networks and Health Inequalities, 35.
6. Karimi, F., & Oliveira, M. (2022). On the inadequacy of nominal assortativity for assessing homophily in networks. arXiv preprint arXiv:2211.10245.

What are social networks?

Social networks consists of **actors** (nodes) and **relations** (edges)

- Actor → person, company, county
 - It may also include entities produced or generated by individuals, e.g., email, retweet.
- Relation → friendship, collaboration, partnership, trade, war, hierarchy
 - When the actor is a resource, the relation may represent a spreading dynamic, access to opportunities, etc.



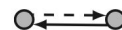
How do networks form?

Social mechanisms in micro-level govern how we organize our social lives and interact with others. Some of those social theories include: group identity theory, cultural evolutions, and broadly social psychology theories.

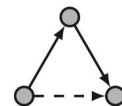
Prevalent **mechanisms** identified in social science literatures:

- Preferential attachment a.k.a. rich-get-richer or Matthew effect
- Homophily and assortative mixing
- Social balance and triadic closure

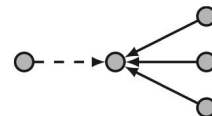
Reciprocity



Transitivity



Popularity



Activity



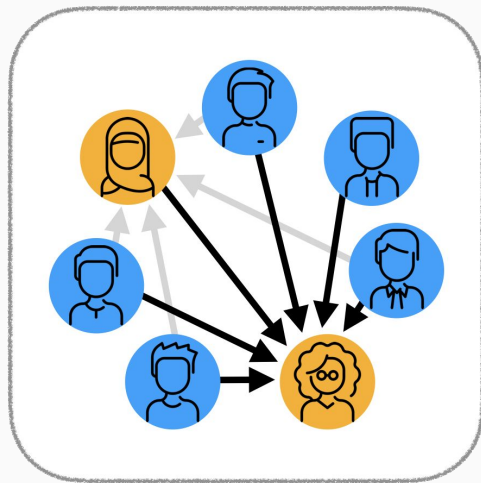
Attraction



Homophily



Covered
in this
tutorial



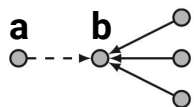
Popularity

The tendency of actors to connect to those who receive ties from many others.

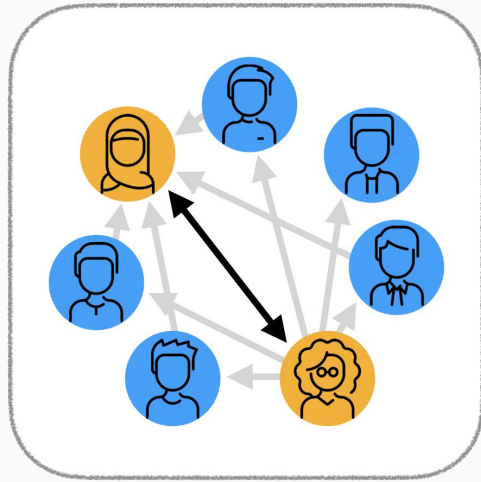
The structural position is defined by the **in-degree of the receiver** (target node) of an explained tie.

- **Matthew effect** mechanism in science discussed by Merton (1968).
Increasing recognition of an actor's scientific work (e.g. number of ties in a citation network).
- **Preferential attachment** in the work of Barabási and Albert (1999).

The presence of a popularity mechanism is not sufficient evidence that a Matthew effect exists in a network, e.g., other actor **attributes**, such as the career stage or gender, might send similar signals that affect the beliefs of others.



$$P(a \rightarrow b) = P(b|a) = p_{ab} = \frac{k_b}{\sum_c k_c}$$



Homophily

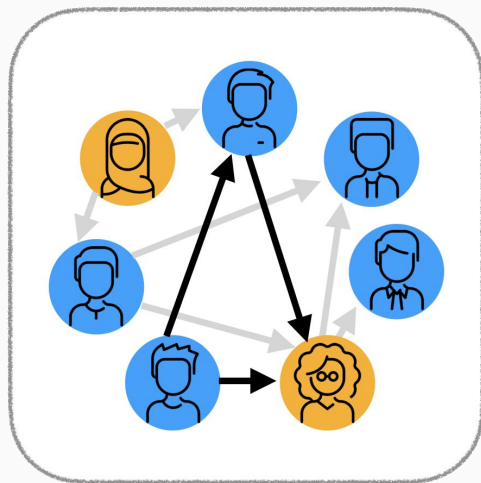
The tendency that similar actors are more likely to connect (Lazarsfeld, et al. 1954; McPherson et al. 2001). Its structural position is defined by **the attribute similarity of the sender and receiver** of the explained tie.

Causal explanations:

- These may be cognitive processes about similarity attraction (Huston and Levinger 1978)
- Structural processes that are affected by existing baseline segregation and social distances in the social setting under study.



$$P(a \rightarrow b) = P(b|a) = p_{ab} = \textit{similarity}(a, b)$$

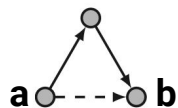


Transitivity

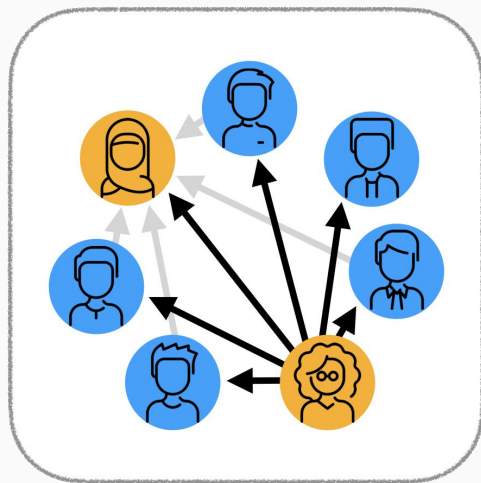
One of the central mechanisms that goes *beyond the dyad* and thus *involves more than two actors*: it expresses the tendency of actors to connect to others to whom they are indirectly tied to through a third actor. The structural position is defined by the presence of a **“two-path” between the sender and the receiver** of the explained tie.

Causal explanations:

- Differences in spatial or social distances and similarity attraction based on an actor attribute (Granovetter 1973).
- Actors may perceive cognitive dissonance and stress if they are not in a positive relationship with those they are indirectly positively tied to (Heider 1958).



$$P(a \rightarrow b) = P(b|a) = p_{ab} = p_{TC} \propto clustering(g)$$



Node Activity

The tendency that actors with specific attributes will be more likely to send ties. The structural positions are defined by **attributes of the sender** of the explained tie. Whether these differences relate to differences in networking resources is not further specified by the mechanism.

a ● --->○ b

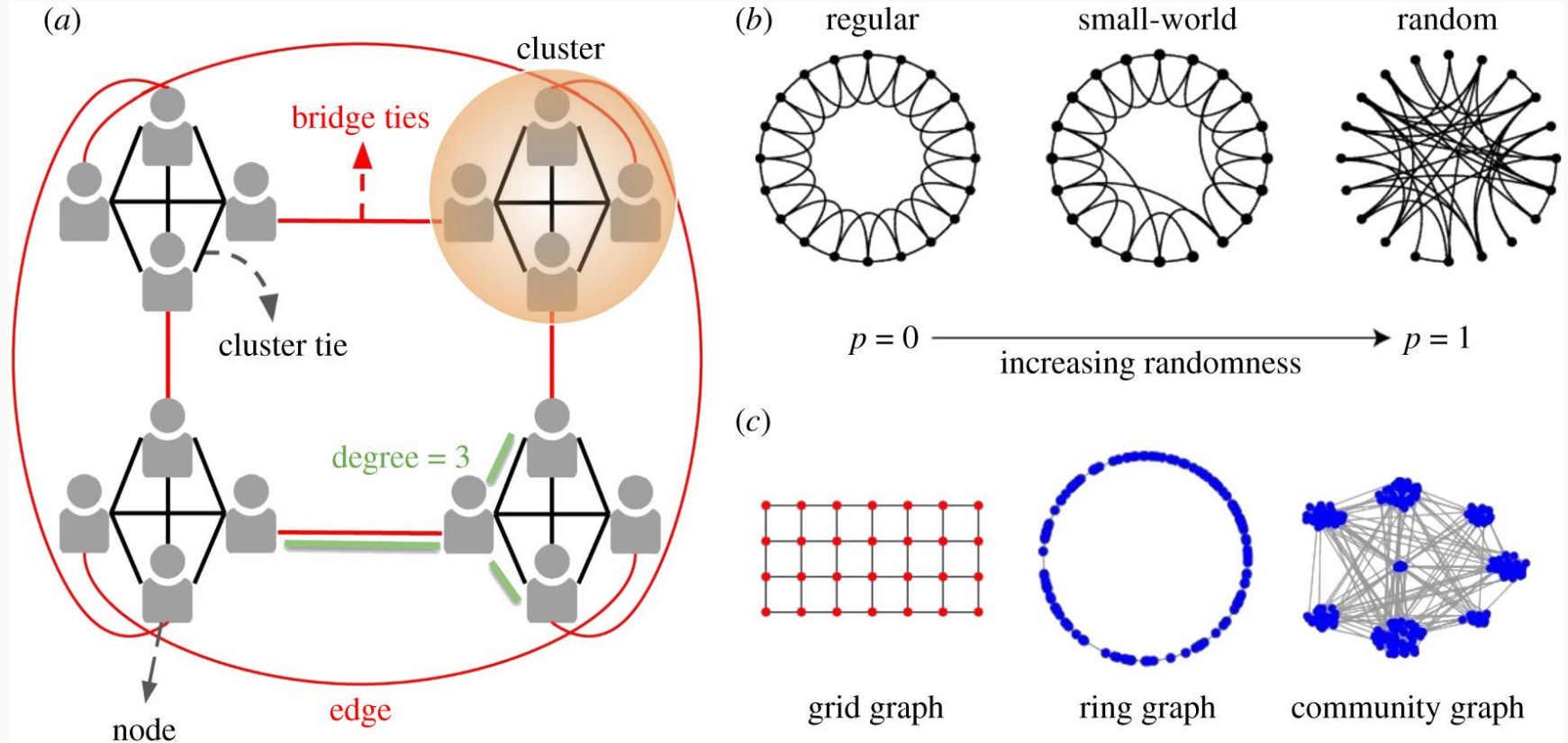
$$P(a \rightarrow b) = P(b|a) = p_{ab} \propto activity(a)$$

Network properties and structure

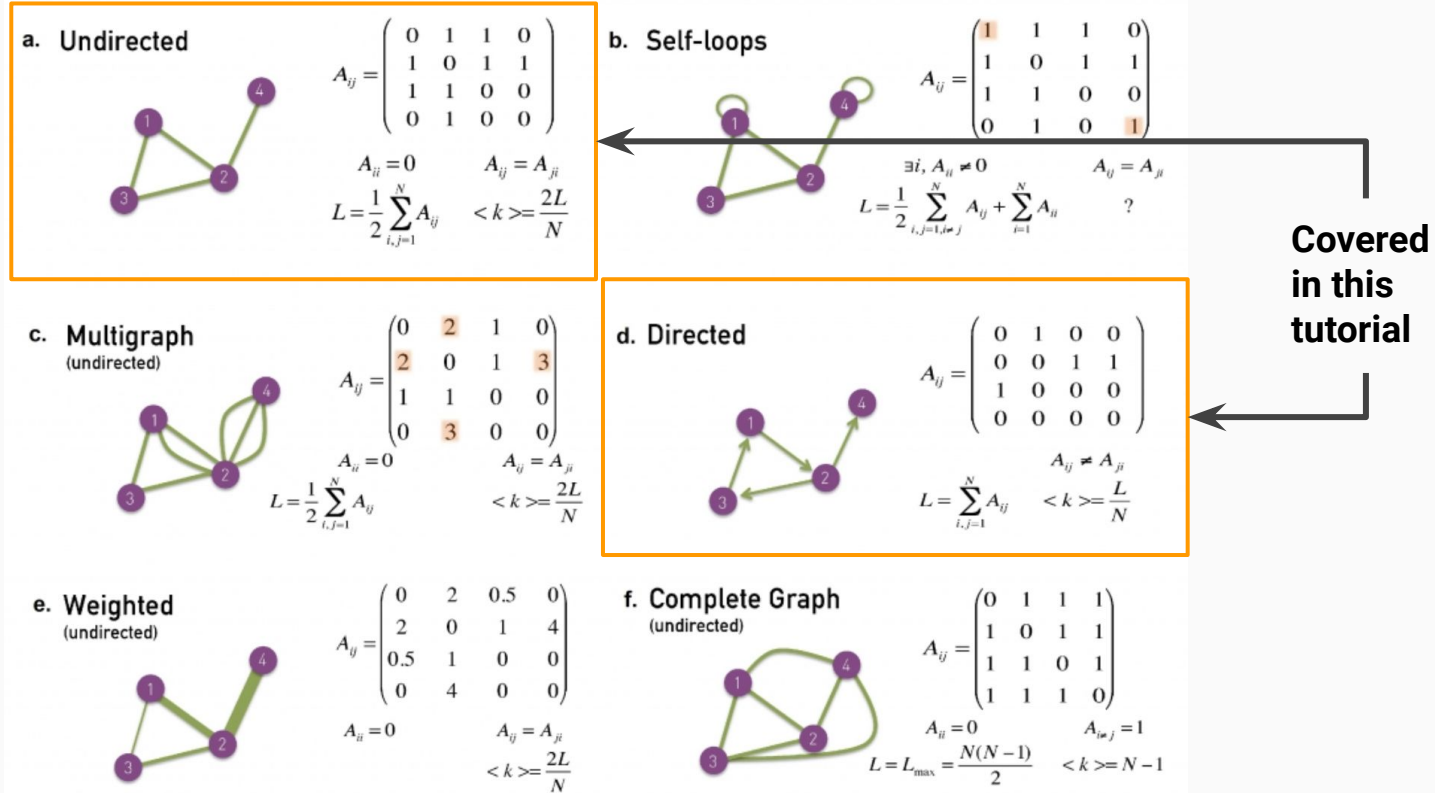
Important properties of social networks

#	Graph	Node	Edge
1	Average and minimum degree	Degree (in/out)	Weights
2	Degree distribution (the probability that a randomly selected node in the network has degree k)	Centrality (e.g., PageRank)	Shortest path
3	Adjacency matrix	Clustering coefficient	Homophilous type (MM, Mm, mm, mM)
4	Sparsity (out of the total possible number of edges, how many actually exist)	Activity (high activity = high outdegree)	
5	Diameter		
6	Average Path Length		
7	Connected components		

Common structure (topology) of social networks



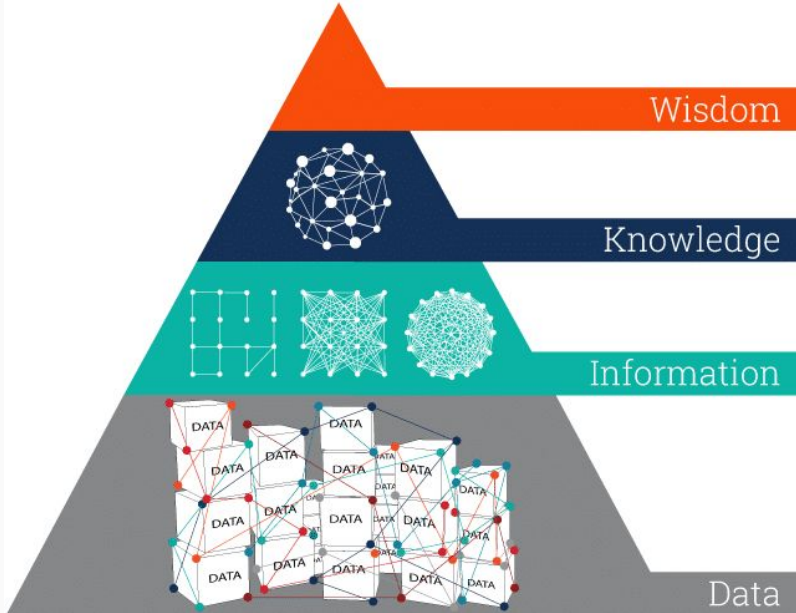
Graphology (elementary property of the underlying graph)



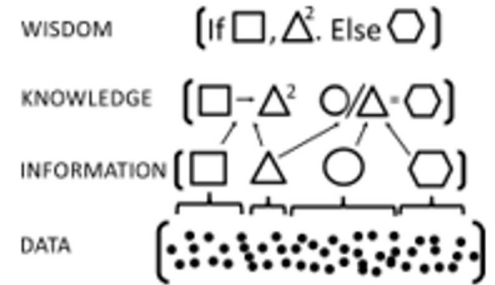
Network models

Why modeling social networks?

The wisdom hierarchy



Each step up
the pyramid
answers
questions
about and
adds value
to the initial data.



The seven uses of Models

1. **Reason:** to identify conditions and deduce logical implications
2. **Explain:** To provide explanations for empirical phenomena
3. **Design:** to choose features of policies, institutions and rules
4. **Communicate:** to relate knowledge and understandings
5. **Act:** to guide policy choices and strategic actions
6. **Predict:** to make numerical and categorical predictions of future and unknown phenomena
7. **Explore:** to investigate possibilities and hypotheticals scenarios

Why modeling social networks?

Here I emphasis on the application of network models in ML and AI draw from our previous research and the below recent survey article

Steinbacher, M., Raddant, M., Karimi, F., Camacho Cuenca, E., Alfarano, S., Iori, G., & Lux, T. (2021). Advances in the agent-based modeling of economic and social behavior. *SN Business & Economics*, 1(7), 99.

Network models

... and where to find them

Class of models in Network Science

**Covered
in this
tutorial**



Model Class	Examples	Characteristics
Static Models	Erdos-Rényi Watts-Strogatz	<ul style="list-style-type: none">• N fixed• p_k exponentially bounded• Static, time independent topologies
Generative Models	Configuration Model Hidden Parameter Model Stochastic Block Model	<ul style="list-style-type: none">• Arbitrary pre-defined p_k• Static, time independent topologies
Evolving Network Models	Barabási-Albert Model Bianconi-Barabási Model Initial Attractiveness Model Internal Links Model Node Deletion Model Accelerated Growth Model Aging Model	<ul style="list-style-type: none">• p_k is determined by the processes that contribute to the network's evolution.• Time-varying network topologies

Table 6.1

Classes of Models in Network Science

The table summarizes the three main modeling frameworks used in network science, together with their distinguishing features.

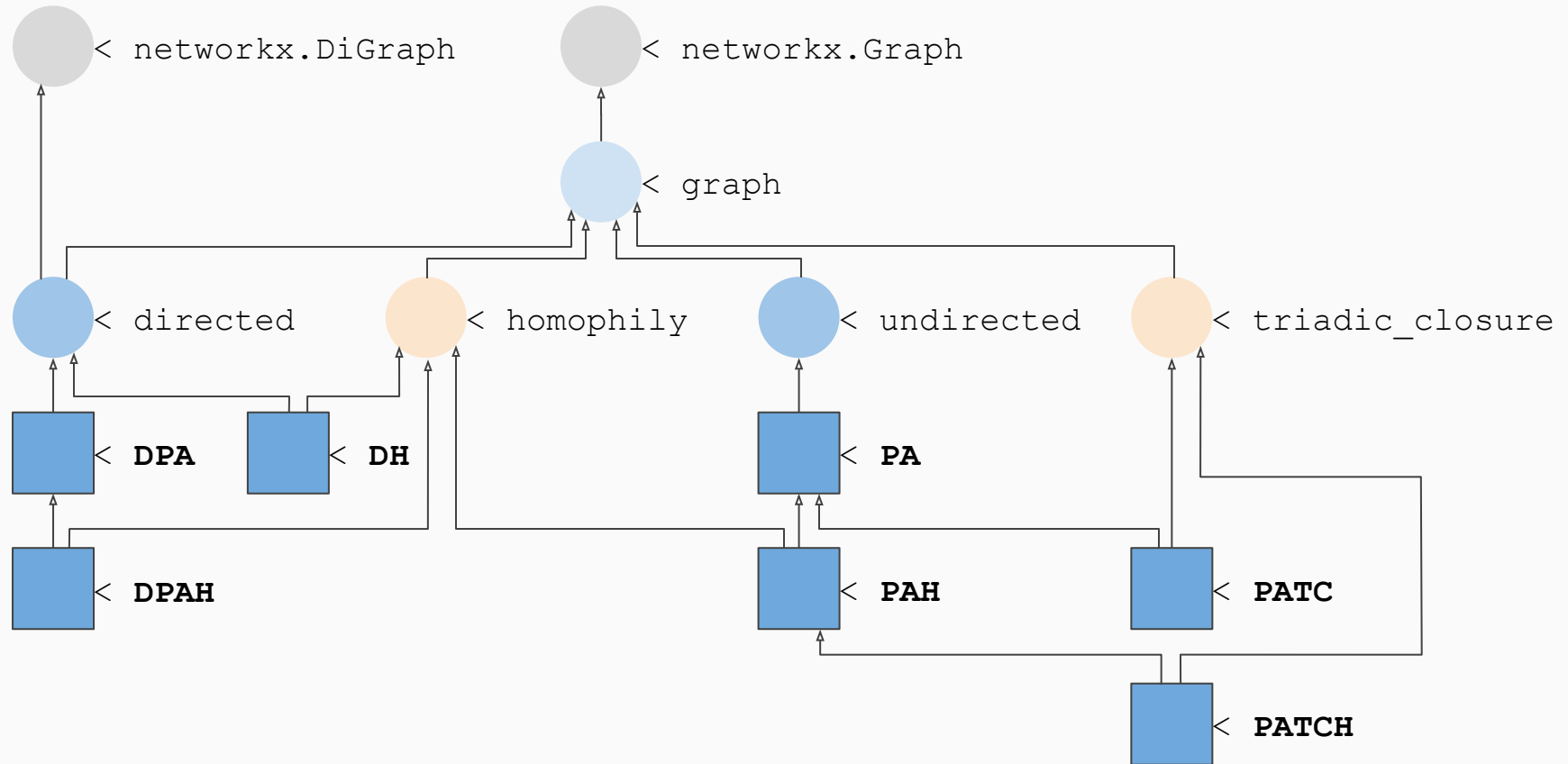
Pre-requisites

Follow these instructions in case you are using your own environment to run the exercises.

Alternatively, you can run all the exercises in the cloud. We have prepared notebooks to run in Google Colab (more details later).

1. Download and install conda
conda.io/projects/conda/en/stable/user-guide/install/download.html
2. Create an environment with python 3.9 or later
`conda create -n "snma" python=3.9 jupyterlab`
3. Activate your newly created conda environment
`conda activate snma`
4. Clone the tutorial in your computer
`git clone`
<https://github.com/snma-tutorial/www2023.git>
5. Install the dependencies
`conda install pip`
`pip install -r requirements.txt`

The `netin` python package (alpha)



Undirected networks

PA

preferential attachment

PATC

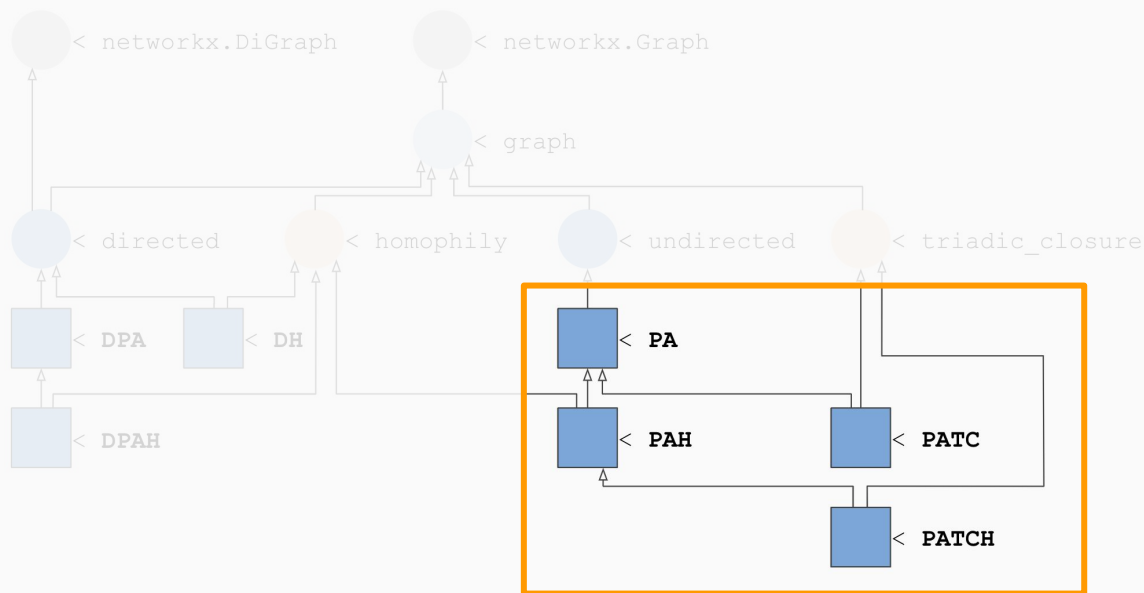
PA + triadic closure

PAH

PA + homophily

PATCH

PAH + triadic closure



```
class netin.PA(...)
```

Creates a new PA instance. An undirected graph with preferential attachment.

Parameters

`n: int`

number of nodes (minimum=2)

`k: int`

minimum degree of nodes (minimum=1)

`f_m: float`

fraction of minorities (minimum=1/n, maximum=(n-1)/n)

`seed: object`

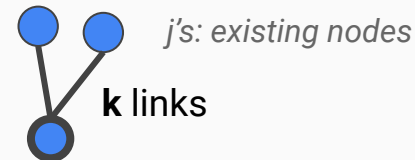
seed for random number generator

STEP 1:

$k = 2$

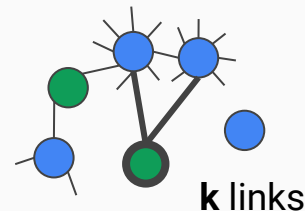


STEP 2:



i: new node (and then existing node)

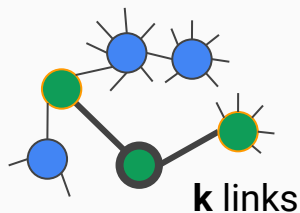
STEP 3 $\leq s \leq n$:



$$p_{ab} = \frac{k_b}{\sum_c k_c}$$


```
class netin.PAH(...)
```

STEP $3 \leq s \leq n$:



$$p_{ab} = \frac{h_{ab}k_b}{\sum_c h_{ac}k_c}$$

Creates a new PAH instance. An undirected graph with preferential attachment and homophily.

Parameters

n: int

number of nodes (minimum=2)

k: int

minimum degree of nodes (minimum=1)

f_m: float

fraction of minorities (minimum=1/n, maximum=(n-1)/n)

h_MM: float

homophily (similarity) between majority nodes (minimum=0, maximum=1.)

h_mm: float

homophily (similarity) between minority nodes (minimum=0, maximum=1.)

seed: object

seed for random number generator

```
class netin.PATC(...)
```

Creates a new PATC instance. An undirected graph with preferential attachment and triadic closure.

Parameters

n: int

number of nodes (minimum=2)

k: int

minimum degree of nodes (minimum=1)

f_m: float

fraction of minorities (minimum=1/n, maximum=(n-1)/n)

tc: float

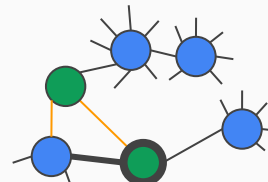
probability of a new edge to close a triad (minimum=0, maximum=1.)

seed: object

seed for random number generator

STEP $3 \leq s \leq n$:
k times

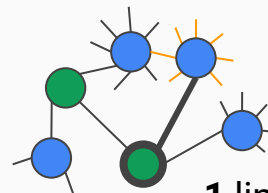
Case 1: p_{TC}



1 link (closes a triangle)

Target 'b' with more nodes in common with 'a'

Case 2: $(1 - p_{TC})$



1 link (via preferential attachment)

$$p_{ab} = \frac{k_b}{\sum_c k_c}$$

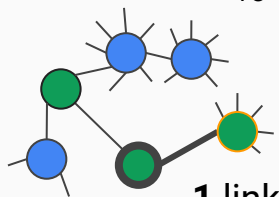
```
class netin.PATCH(...)
```

STEP 3 $\leq s \leq n$:

k times

Case 1: p_{TC}

Case 2: $(1 - p_{TC})$



1 link

(via preferential attachment and homophily)

$$p_{ab} = \frac{h_{ab}k_b}{\sum_c h_{ac}k_c}$$

Creates a new PATCH instance. An undirected graph with preferential attachment, homophily, and triadic closure.

Parameters

n: int

number of nodes (minimum=2)

k: int

minimum degree of nodes (minimum=1)

f_m: float

fraction of minorities (minimum=1/n, maximum=(n-1)/n)

h_MM: float

homophily (similarity) between majority nodes (minimum=0, maximum=1.)

h_mm: float

homophily (similarity) between minority nodes (minimum=0, maximum=1.)

tc: float

probability of a new edge to close a triad (minimum=0, maximum=1.)

- F. Karimi, M. Génois, C. Wagner, P. Singer, & M. Strohmaier "Homophily influences ranking of minorities in social networks" Scientific reports 2018
- P. Holme and B. J. Kim, "Growing scale-free networks with tunable clustering" Phys. Rev. E 2002.
- PATCH: Forthcoming...

Directed networks

DPA

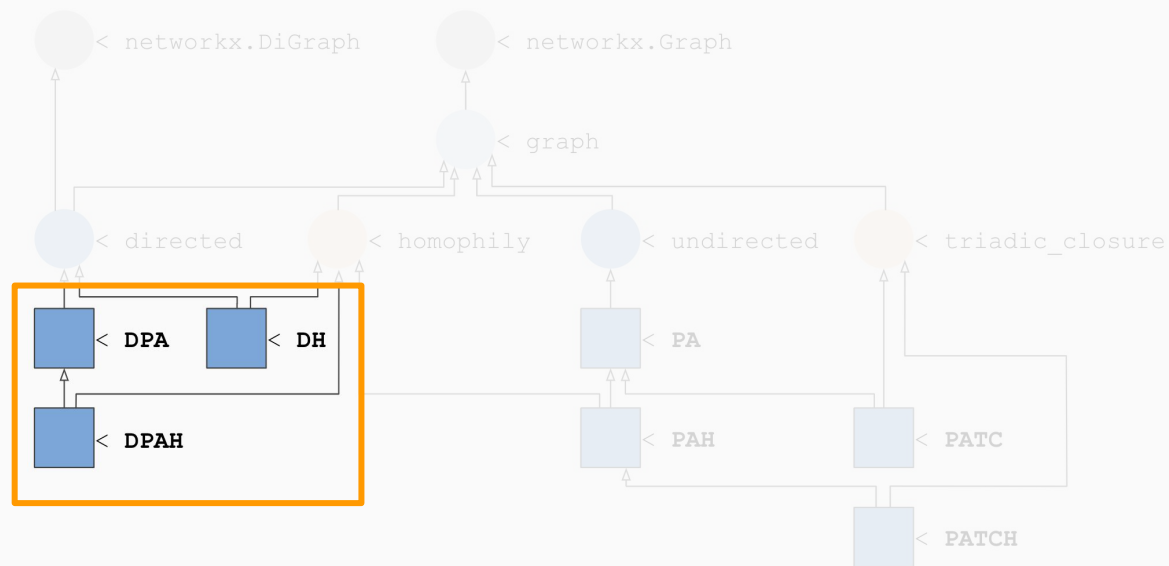
preferential attachment

DH

homophily

DPAH

DPA + homophily



```
class netin.DPA(...)
```

Do while density < d:

Step 1: choose a **source** node
(using the activity of all nodes)

Step 2: choose a **target** node *
(using preferential attachment \rightarrow in_degree)

* target nodes must have out_degree > 0
(should have been picked as source at least once)

Creates a new DPA instance. A directed graph with preferential attachment.

Parameters

n: int

number of nodes (minimum=2)

d: float

edge density (minimum=0, maximum=1)

f_m: float

fraction of minorities (minimum=1/n, maximum=(n-1)/n)

plo_M: float

activity (out-degree power law exponent) majority group (minimum=1)

plo_m: float

activity (out-degree power law exponent) minority group (minimum=1)

seed: object

seed for random number generator

```
class netin.DH(...)
```

Do while density < d:

Step 1: choose a **source** node

(using the activity of all nodes)

Step 2: choose a **target** node *

(using mixing matrix → homophily within/across groups)

* target nodes must have out_degree > 0

(should have been picked as source at least once)

Creates a new DH instance. A directed graph with homophily.

Parameters

n: int

number of nodes (minimum=2)

d: float

edge density (minimum=0, maximum=1)

f_m: float

fraction of minorities (minimum=1/n, maximum=(n-1)/n)

plo_M: float

activity (out-degree power law exponent) majority group (minimum=1)

plo_m: float

activity (out-degree power law exponent) minority group (minimum=1)

h_MM: float

homophily within majority group (minimum=0, maximum=1)

h_mm: float

homophily within minority group (minimum=0, maximum=1)

seed: object

seed for random number generator

```
class netin.DPAH(...)
```

Do while density < d:

Step 1: choose a **source** node
(using the activity of all nodes)

Step 2: choose a **target** node *
(using mixing matrix and preferential attachment)

* target nodes must have out_degree > 0
(should have been picked as source at least once)

Creates a new DPAH instance. A directed graph with preferential attachment and homophily.

Parameters

n: int
number of nodes (minimum=2)

d: float
edge density (minimum=0, maximum=1)

f_m: float
fraction of minorities (minimum=1/n, maximum=(n-1)/n)

plo_M: float
activity (out-degree power law exponent) majority group (minimum=1)

plo_m: float
activity (out-degree power law exponent) minority group (minimum=1)

h_MM: float
homophily within majority group (minimum=0, maximum=1)

h_mm: float
homophily within minority group (minimum=0, maximum=1)

seed: object
seed for random number generator

Exercise

Task:

1. Generate 3 undirected graphs.
2. Get to know them (attributes, visualize them, etc.)
3. Plot and compare their type of edges (MM, Mm, mm, mM).
4. Plot and compare the pdf of their degree distributions.

Repeat for other graphs and check the effects of other mechanisms.

(30 min)

Open `1_exercise.ipynb`

1. Alternatively, you can open the notebook from Google Colab (you need a Google account):

bit.ly/snma2023-notebooks

Lunch break
see you back at **13:30**