Social network modeling and applications

Tutors:
Lisette Espín-Noboa, Tiago Peixoto, and Fariba Karimi

10am - 5pm

bit.ly/snma2023

Who are we?

COMPLEXITY
SCIENCE
HUB
VIENNA

CENTRAL
EUROPEAN
UNIVERSITY

TECHNISCHE
UNIVERSITÄT
WIEN

Fariba Karimi

- **Assistant Professor**
  Vienna University of Technology (TU Wien)
- **Team lead**
  Complexity Science Hub Vienna
- **PhD. in Physics**
  Umea University
- **Research**
  Network inequalities, computational social science

www.networkinequality.com

Tiago Peixoto

- **Associate Professor**
  Central European University (CEU)

- **PhD. in Physics**
  University of São Paulo

- **Research**
  Complex networks, network inference, Bayesian modeling

  skewed.de

Lisette Espín-Noboa

- **Postdoc**
  Complexity Science Hub Vienna (CSH) & Central European University (CEU)

- **PhD. in Computer Science**
  University of Koblenz-Landau

- **Research**
  Network fairness, social network analysis, algorithmic auditing, computational social science

www.lisetteespin.info

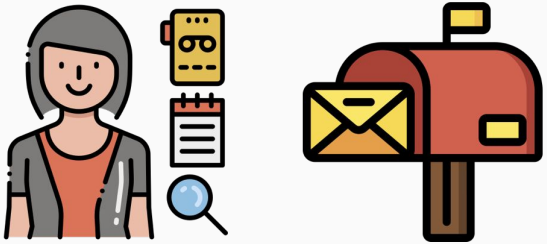
Why do we need this tutorial?

Because
social networks
are everywhere!

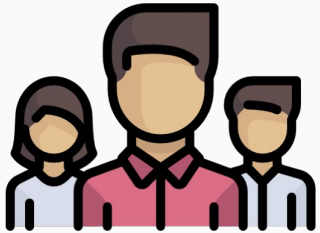
image source: https://insights.dice.com/2019/09/25/networking-building-contacts-effective

Social networks in the era of big data

Before



Field observations and surveys



"Designed" data covering few people in small geographical areas

Now

Digital footprints from social media, phones, online surveys

"Organic" data covering almost the entire world

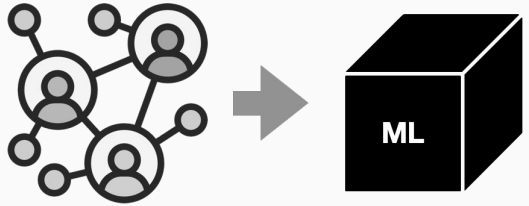
Social networks in the era of big data and machine learning

Problem #1

Traditional methods do not scale

We need new tools to characterize edge formation

Problem #2



ML algorithms are not transparent

classification

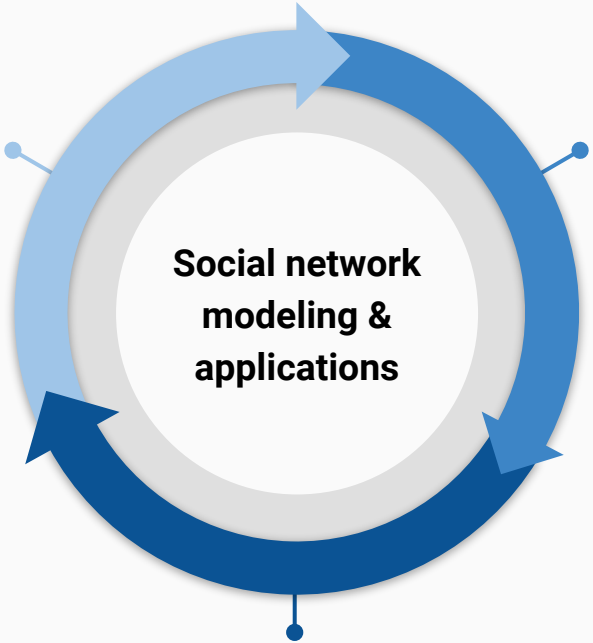
ranking

ML algorithms need to be interpretable and explainable

What we will cover in this tutorial

PARADIGM 1

From social theories to models
(e.g., homophily, preferential attachment)



PARADIGM 2

From data to models
(e.g., model fitting, inference)

APPLICATIONS

Network inequalities & algorithmic bias
(i.e., sampling and ranking)

What you will learn today

1	**How to generate more realistic synthetic graphs**	• Useful when real networks are too big, and for data protection • How? Using `netin` and `graph-tool`
2	**How to study social phenomena when no real network is available**	• How do networks (edges) form? • How to disentangle homophily and triadic closure
3	**To audit network-based algorithms**	• What-if scenarios via simulations • To audit network-based algorithms ○ When does my model fail?

What this tutorial is NOT about

- Extensive review of all existing random network models
  - Here we focus on models that help replicating the most important properties and mechanisms found in real-world social networks such as preferential attachment, homophily, and triadic closure.
- Extensive review of network-based algorithms
  - Here we show the main ingredients of how to use synthetic data to audit your own algorithm. Due to time limitations we will cover only sampling biases and ranking inequalities. But the same logic applies to any other algorithm.
- `graph-tool` “101”
  - Exercise #2 covers model inference and selection using **`graph-tool`**, however, participants should know already the basic concepts and syntax of this package. Participants should familiarize with **`graph-tool`** by reading the quick start guide: https://graph-tool.skewed.de/static/doc/quickstart.html

Agenda

Sunday, April 30

10:00 - 17:00 (CDT)

AT&T Hotel and Conference Center
Classroom #103

10:30 - 12:30	Paradigm 1: From social theories to models Tutor: Fariba Karimi • Social theories • Network properties and structure • Network models
12:30 - 13:30	Lunch break
13:30 - 15:30	Paradigm 2: From data to models Tutor: Tiago Peixoto • Model fitting / inference • Model selection • Disentangling homophily & triadic closure
15:30 - 16:40	Applications Tutor: Lisette Espín-Noboa • Biases in sampling subgraphs • Ranking inequalities
16:40 - 17:00	Challenges & open questions Tutor: Lisette Espín-Noboa

Exercises

Please make sure your python environment is ready to go!

Note that if you prefer to run the exercises directly from Google Colab you can skip this info, but recall that you need a Google account.

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 graph-tool
   ```
   conda install -c conda-forge graph-tool
   ```
6. Install the additional dependencies
   ```
   conda install pip
   pip install -r requirements.txt
   ```

Material

All required information is on the tutorial's website:

https://bit.ly/snma2023