10th International Conference on Computational Social Science

Network Diffusion — Framework to Simulate Spreading Processes in Complex Networks

Michał Czuba [1], Mateusz Nurek [1], Damian Serwata [1], Yu-Xuan Qi [2], Mingshan Jia [2], Katarzyna Musial [2], Radosław Michalski [1], Piotr Bródka [1]

[1]Wrocław University of Science and Technology
[2]University of Technology Sydney

20.07.2024

Network Science Lab

Agenda

Network Science Lab

Motivation

Spreading phenomena are one of the problems considered by a network science. They can be obeserved in various areas like: dynamics of political opinions, marketing campaigns, spread of epidemics, computer viruses, etc.

Figure: Artistic representation of a social network.[1]

[1] Source: www.uniroma3.it/articoli/seminario-biased-opinion-dynamics-when-the-devil-is-in-the-details-138122

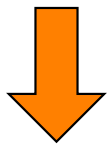
Network Science Lab

Motivation

Analytical approaches to such phenomena are often insufficient for large graphs, prompting researchers to use computational methods, i.e. simulations.

Motivation

Analytical approaches to such phenomena are often insufficient for large graphs, prompting researchers to use computational methods, i.e. simulations.



Thus, many tools were developed to address that issue, allow researchers to get rid of the need to start their experiments from scratch, and enhance the reproducibility of results.

Network Science Lab

Motivation

Here is a bunch of tools that help in simulating diffusion processes in networks:

- **NDlib**[5],
- GLEaMviz[2],
- SimInf[6],
- STEM[3],
- EpiModel[4],
- Sispread[1],
- ...

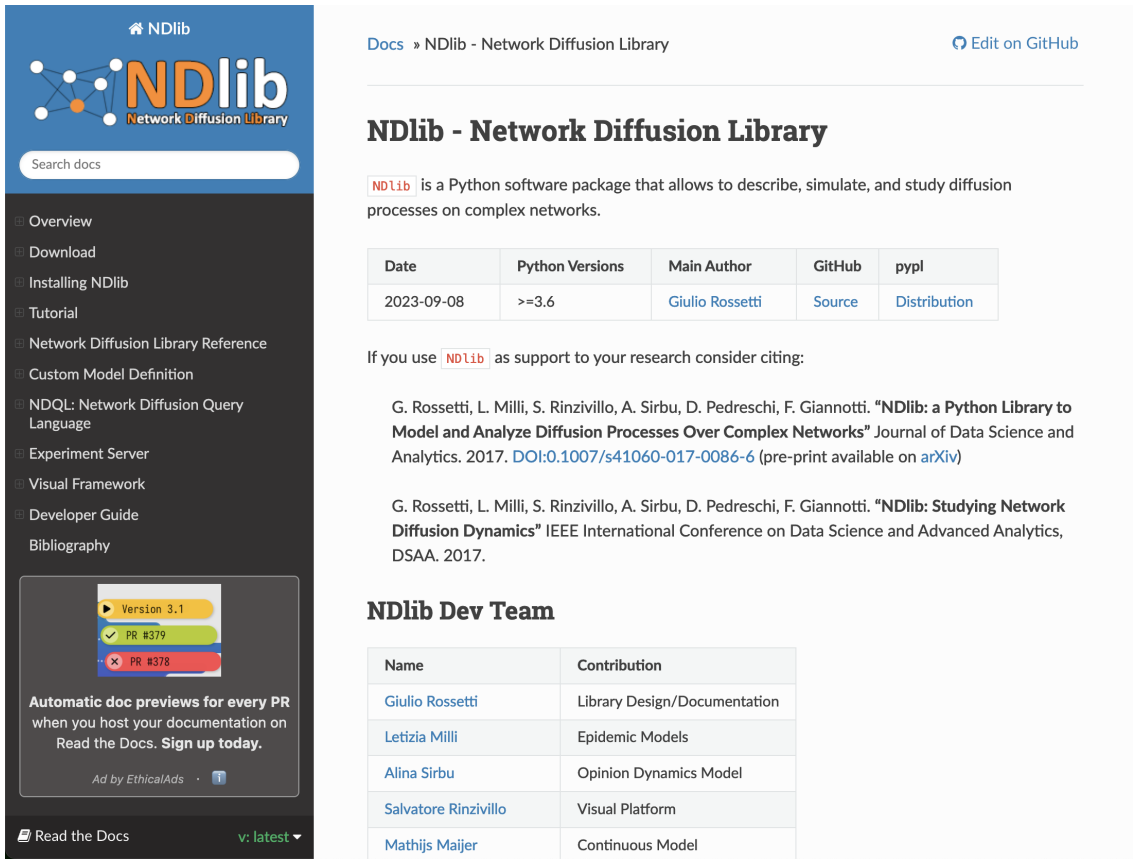


Figure: NDlib's website.

Network Science Lab

Motivation

However, if we consider more complex...

...network models (e.g. multilayer graphs)...

...diffusion models (e.g. spreading of coexisting processes)...

...**a gap among the available toolkits** emerges.

Network Science Lab

Motivation

For instance, which library should we choose if we have to * ?

* model simultaneous spreading of opinions about advertised products
* convert dataset of interactions into a temporal graph
* process a multilayer network to extract its centrality measures
* identify super spreaders in a multiplex network under ICM

Network Science Lab

Motivation

My research group works on problems related to multilayer/temporal networks, spreading phenomena, influence maximisation, data streams, etc.

Motivation

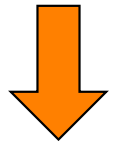
My research group works on problems related to multilayer/temporal networks, spreading phenomena, influence maximisation, data streams, etc.

During the research, we produced a code that can be reused as a boilerplate for further experiments, with interfaces allowing anyone to quickly implement a custom spreading model.

Network Science Lab

Motivation

My research group works on problems related to multilayer/temporal networks, spreading phenomena, influence maximisation, data streams, etc.



During the research, we produced a code that can be reused as a boilerplate for further experiments, with interfaces allowing anyone to quickly implement a custom spreading model.

With regard to the observed gap among existing toolkits, we decided to merge and wrap the code into a library to share it with the community as a small contribution to make experiments easier.

Network Science Lab

Key Features of `network-diffusion`

Functionalities of the library:

- end-to-end workflow for simulations of spreading phenomena,

Network Science Lab

Key Features of `network-diffusion`

Functionalities of the library:

- end-to-end workflow for simulations of spreading phenomena,
- predefined (basic) spreading models,

Network Science Lab

Key Features of `network-diffusion`

Functionalities of the library:

- end-to-end workflow for simulations of spreading phenomena,
- predefined (basic) spreading models,
- an interface for implementing custom discrete spreading models,

Network Science Lab

Key Features of `network-diffusion`

Functionalities of the library:

- end-to-end workflow for simulations of spreading phenomena,
- predefined (basic) spreading models,
- an interface for implementing custom discrete spreading models,
- support for temporal networks (CogSNet and snapshot-based),

Network Science Lab

Key Features of `network-diffusion`

Functionalities of the library:

- end-to-end workflow for simulations of spreading phenomena,
- predefined (basic) spreading models,
- an interface for implementing custom discrete spreading models,
- support for temporal networks (CogSNet and snapshot-based),
- support for the multilayer networks,

Network Science Lab

Key Features of `network-diffusion`

Functionalities of the library:

- end-to-end workflow for simulations of spreading phenomena,
- predefined (basic) spreading models,
- an interface for implementing custom discrete spreading models,
- support for temporal networks (CogSNet and snapshot-based),
- support for the multilayer networks,
- most of centrality metrics extended to multilayer networks.

Network Science Lab

Key Features of `network-diffusion`

Environmental requirements/features:

- released under MIT licence,
- support for Linux, macOS, and Windows[2],
- Python >= 3.10 compatibility,
- C snippets in the CogSNet module to speed-up computations,
- NetworkX compatibility.

[2] Although we develop and use it mostly on Unix OSs

Network Science Lab

Example I - Predefined Model

In this example, we will see how to trigger spread under the Linear Threshold Model in a toy, multilayer network.

Linear Threshold Model

Each node:

- can fall in two states: *active* and *inactive*,
- becomes *active* if the fraction of its *active* neighbors to all neighbours exceeds certain threshold.

In case of multilayer networks the actors not the nodes[3] are a subject of the diffusion. Thus, we have to define how to aggregate impulses from the layers. In this example we will consider "OR" strategy.

[3] which can be considered as avatars of the actors on the network's layers

Network Science Lab

Example I - Predefined Model

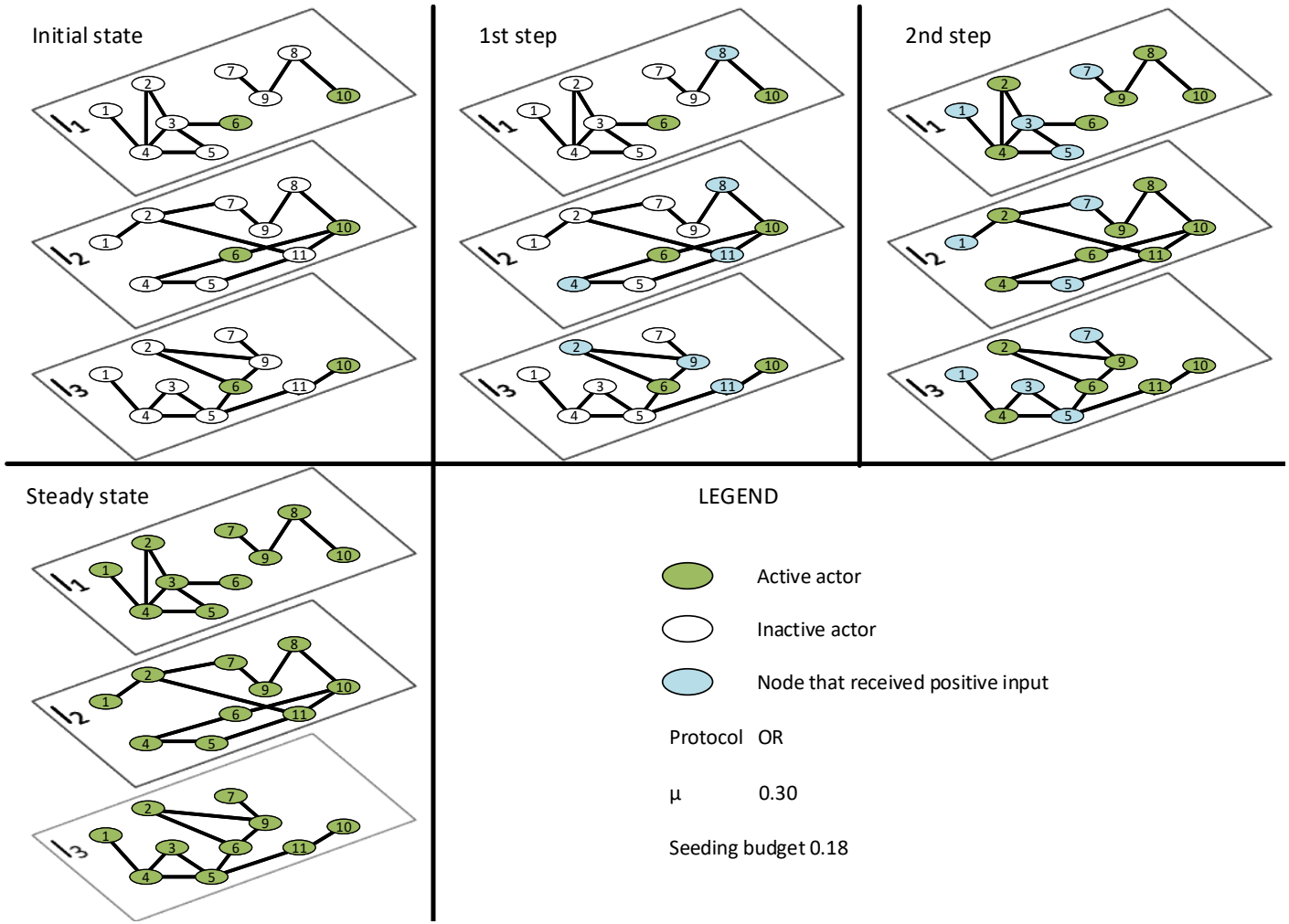


Figure: Propagation according to LTM with the *OR* strategy in a multilayer network - active actors: seeds $\{6, 10\}$, in a stable state: all of them.

Network Science Lab

Example I - Predefined Model

Let's model this problem with `network-diffusion`!

Example II - Custom Model

In this example we will consider a joint disease-awareness model (SIR-UA) that can be used e.g. to assess the effectiveness of various countermeasures against the spread of COVID-19:

Table: Transition weights with explanation.

Symbol	Description
α	probability of infection for unaware agents
α'	probability of infection for aware agents
β	probability of recovery
γ	probability of awareness for uninfected agents
δ	probability of awareness for infected agents

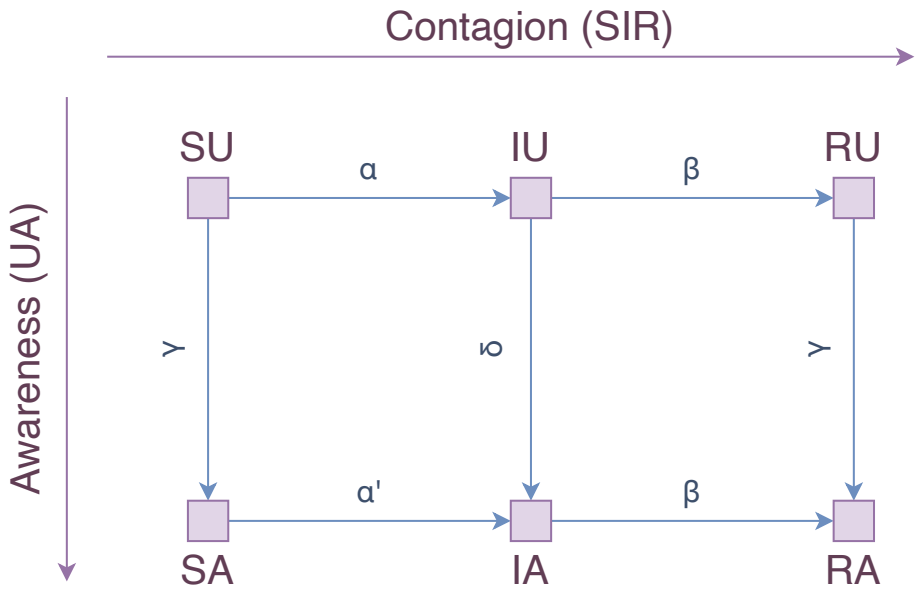


Figure: State and transition graph for SIR-UA.

Network Science Lab

Example II - Custom Model

To create an own spreading model, we have to extend the abstract base class (`nd.models.BaseModel`) by implementing:

- a field `_compartmental_graph`,
- a field `_seed_selector`,
- a method `determine_initial_states`,
- a method `agent_evaluation_step`,
- a method `network_evaluation_step`,
- methods `__str__` and `get_allowed_states`.

The following entities (with implemented method `update_network`) are utilised in the class `nd.Simulator` which orchestrates the experiment.

Network Science Lab

Example II - Custom Model

Let's model this problem with `network-diffusion`!

Resources and References

The library can be installed with:

```
pip install network-diffusion
```

We have also published other useful resources:

- PyPI website: `pypi.org/project/network-diffusion`,

Network Science Lab

Resources and References

The library can be installed with:

```
pip install network-diffusion
```

We have also published other useful resources:

- PyPI website: `pypi.org/project/network-diffusion`,
- GitHub page: `github.com/anty-filidor/network_diffusion`,

Network Science Lab

Resources and References

The library can be installed with:

```
pip install network-diffusion
```

We have also published other useful resources:

- PyPI website: `pypi.org/project/network-diffusion`,
- GitHub page: `github.com/anty-filidor/network_diffusion`,
- Reference guide: `network-diffusion.readthedocs.io`,

Network Science Lab

Resources and References

The library can be installed with:

```
pip install network-diffusion
```

We have also published other useful resources:

- PyPI website: `pypi.org/project/network-diffusion`,
- GitHub page: `github.com/anty-filidor/network_diffusion`,
- Reference guide: `network-diffusion.readthedocs.io`,
- Paper with more use-cases presented: `arxiv.org/abs/2405.18085`.

Network Science Lab

Limitations of `network-diffusion`

There is still some work to do...

- computational performance can be better,
- we are limited to discrete spreading models,
- we cannot model agent's complex states (e.g. internal/external),
- much less pre-defined spreading models than in NDlib,
- library maintenance guaranteed up to my graduation ☺.

Network Science Lab

References I

[1] Fabian Alvarez et al. "Sispread: A Software to Simulate Infectious Diseases Spreading on Contact Networks". In: Methods of information in medicine 46 (2007), p. 19. DOI: 10.1055/s-0038-1627827. URL: https://doi.org/10.1055/s-0038-1627827.

[2] Wouter Van den Broeck et al. "The GLEaMviz Computational Tool, a Publicly Available Software to Explore Realistic Epidemic Spreading Scenarios at the Global Scale". In: BMC Infectious Diseases 11.1 (Feb. 2011), p. 37. ISSN: 1471-2334. DOI: 10.1186/1471-2334-11-37. URL: https://doi.org/10.1186/1471-2334-11-37.

[3] Judith Douglas et al. "STEM: An Open Source Tool for Disease Modeling". In: Health security (Aug. 2019). DOI: 10.1089/hs.2019.0018. URL: https://doi.org/10.1089/hs.2019.0018.

Network Science Lab

References II

[4] Samuel M Jenness, Steven M Goodreau, and Martina Morris. "EpiModel: an R package for mathematical modeling of infectious disease over networks". In: Journal of statistical software 84 (2018). DOI: 10.18637/jss.v084.i08. URL: https://doi.org/10.18637/jss.v084.i08.

[5] Giulio Rossetti et al. "NDlib: A Python Library to Model and Analyze Diffusion Processes Over Complex Networks". In: Int. J. Data Sci. Anal. 5.1 (Feb. 2018), pp. 61–79. ISSN: 2364-4168. DOI: 10.1007/s41060-017-0086-6. URL: https://doi.org/10.1007/s41060-017-0086-6.

[6] Stefan Widgren et al. "SimInf: An R Package for Data-Driven Stochastic Disease Spread Simulations". In: Journal of Statistical Software 91.12 (2019), pp. 1–42. DOI: 10.18637/jss.v091.i12. URL: https://doi.org/10.18637/jss.v091.i12.

Network Science Lab

Thank you for your attention!

Scan this QR code to reach the main page of the project ☺

Network Science Lab