Network Diffusion — Framework to Simulate Spreading Processes in Complex Networks

Wrocław University of Science and Technology

Network Science Lab

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

Introduction

The problem of selecting an optimal seed set to maximise influence in networks has been a subject of intense research in recent years. However, there is still a missing part to be tackled: **multilayer networks**. Methods robust for one-layer-graphs are not easily applicable to their multilayer counterparts. That narrows their usability in real-case scenarios such as marketing campaigns, misinformation tracking, or epidemiology, where multilayer networks usually express actual conditions better. **In this work, we show the efficiency of various metrics used to determine the initial seed set for the Multilayer Linear Threshold Model** (MLTM).

Extending the LTM to multilayer networks

Linear Threshold Model in its initial form [1] cannot be directly applied to multilayer networks — **actors are the subject of the process, while the nodes are their auxiliary representation**... Therefore, we need to define:

- what does it mean that an actor is (or is not) active,
- how does it relate to diffusion dynamics taking place within layers, where it is represented.

In our research, we used the approach proposed by [2] with amendments so that a homogeneity among actors has been imposed in the sense of an activation threshold (μ) and a protocol (v.i.).

Protocol function in MLTM

According to [2], state of the actor n of a multilayer network $M = (N, L, V, E)$ in the time step t is determined by a following function:

$$x_n(t) = \begin{cases} 1, & \text{if } \ y_n(t) \geq \delta \text{ or } \ x_n(t-1) = 1 \\ 0, & \text{otherwise} \end{cases}$$

Where:

δ - a parameter of the model, $\delta \in [\frac{1}{|L|}, 1]$,

$y_n(t)$ - a mean input of actor n (represented in K layers) in time t, $y_n(t) = |K|^{-1} \sum_{k \in K} y_v^k(t)$.

$y_v^k(t)$ - an impulse of node v from layer k in time t, $y_v^k(t) \in \{0, 1\}$

Toy example

We decided to examine two extreme cases: $\delta = \mathbf{1}$ **(AND) and delta** $= \frac{1}{L}$ **(OR)**. In the former one, an actor gets activated if it receives sufficient influence on all layers where it is represented, and conversely the latter, where sufficient input in at least one layer is enough for activation.

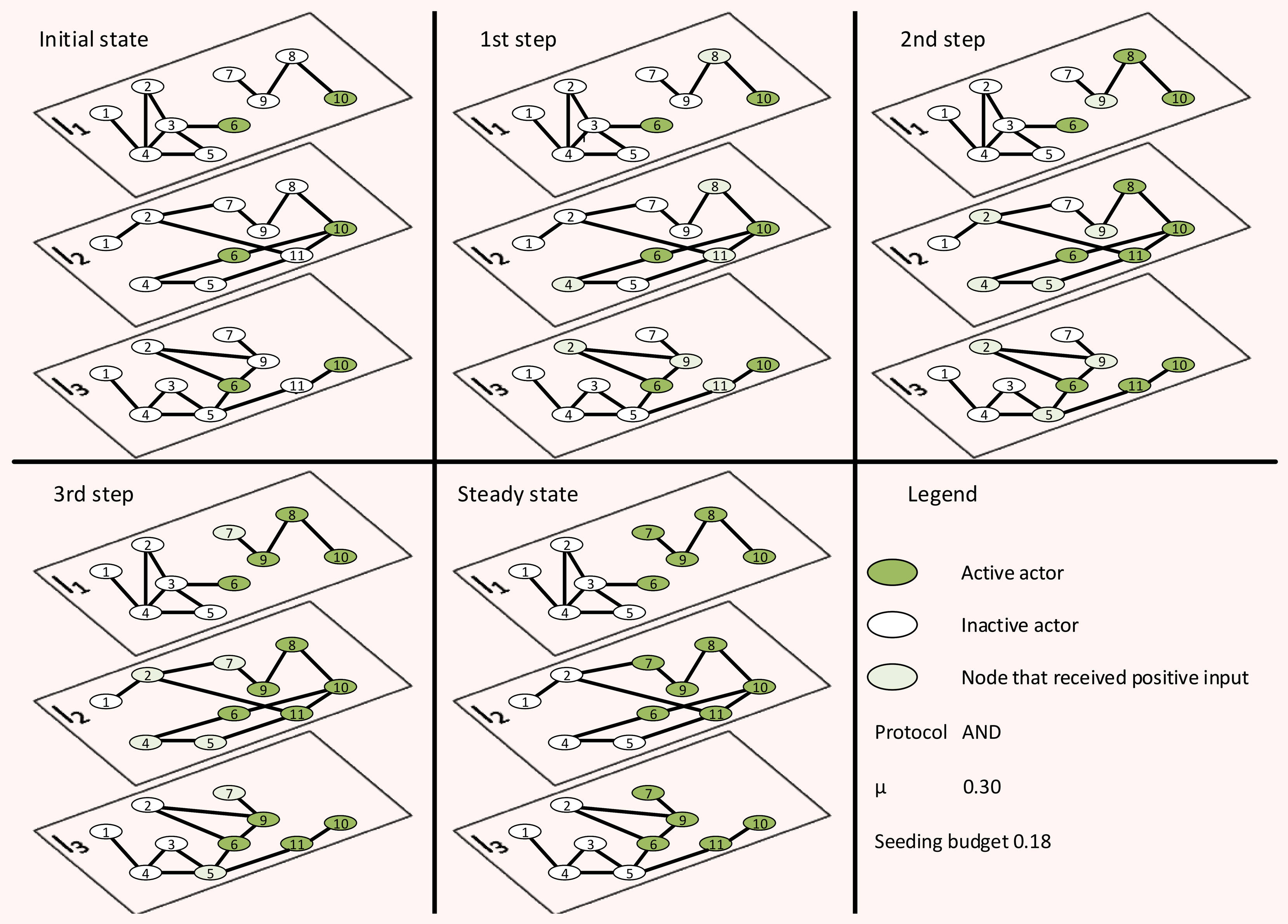

Figure 1. Example of spreading of MLTM in toy network with protocol AND.

A problem we tackled

Budget constrained influence maximisation

Key Features

- **End-to-End Simulation Workflow**: The library enables users to simulate diffusion processes in complex networks with ease. Whether you are studying information spread, disease propagation, or any other diffusion phenomena, this library has you covered.
- **Support for Temporal Network Models**: You can work with temporal models, allowing you to capture the dynamics of processes over time. These temporal models can be created using regular time windows or leverage **CogSnet**.
- **Support for Multilayer Network Models**: The library supports multilayer networks, which are essential for modelling real-world systems with interconnected layers of complexity
- **Predefined Spreading Models**: You have the option to use predefined diffusion models such as the Linear Threshold Model, Independent Cascade Model, and more. Those are implemented to simplify the simulation process, allowing users to focus on their specific research questions.
- **An Interface for Implementing Custom Spreading Models**: Additionally, `network-diffusion` allows you to define your own diffusion models using open interfaces, providing flexibility for researchers to tailor simulations to their unique requirements.
- **New Centrality Measures**: The library provides a wide range of centrality measures specifically designed for multilayer networks. These measures can be valuable for selecting influential seed nodes in diffusion processes.
- **NetworkX Compatibility**: The package is built on top of `networkx`, ensuring seamless compatibility with this popular Python library for network analysis. You can easily integrate it into your existing `networkx`-based workflows.

A Short Example

```python
import network_diffusion as nd

# define the model with its internal parameters
spreading_model = nd.models.MICModel(
    seeding_budget=[90, 10, 0],  # 95% act suspected, 10% infected, 0% recovered
    seed_selector=nd.seeding.RandomSeedSelector(),  # pick infected act randomly
    protocol="OR",  # how to aggregate impulses from the network's layers
    probability=0.5,  # probability of infection
)

# get the graph - a medium for spreading
network = nd.mln.functions.get_toy_network_piotr()

# perform the simulation that lasts four epochs
simulator = nd.Simulator(model=spreading_model, network=network)
logs = simulator.perform_propagation(n_epochs=3)

# obtain detailed logs for each actor in the form of JSON
raw_logs_json = logs.get_detailed_logs()

# or obtain aggregated logs for each of the network's layer
aggregated_logs_json = logs.get_aggregated_logs()

# or just save a summary of the experiment with all the experiment's details
logs.report(visualisation=True, path="my_experiment")
```

Try `network-diffusion`

- we confirmed that efficiency of centrality based heuristics is located between *random* and *greedy*,
- we have proposed an extension of VoteRank, PageRank and k-shell to multilayer networks,
- for both protocols VoteRank is the most efficient method,
- in real case scenarios, it could be better to make the μ smaller (i.e. invest in a mass marketing) than to put efforts in selecting the optimal S_0 (i.e. conduct the campaign with influencers).

References

[1] David Kempe, Jon Kleinberg, and Éva Tardos. "Maximizing the Spread of Influence Through a Social Network". In: *9th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2003, p. 137.

[2] Yaofeng Desmond Zhong, Vaibhav Srivastava, and Naomi Ehrich Leonard. "Influence Spread in the Heterogeneous Multiplex Linear Threshold Model". In: