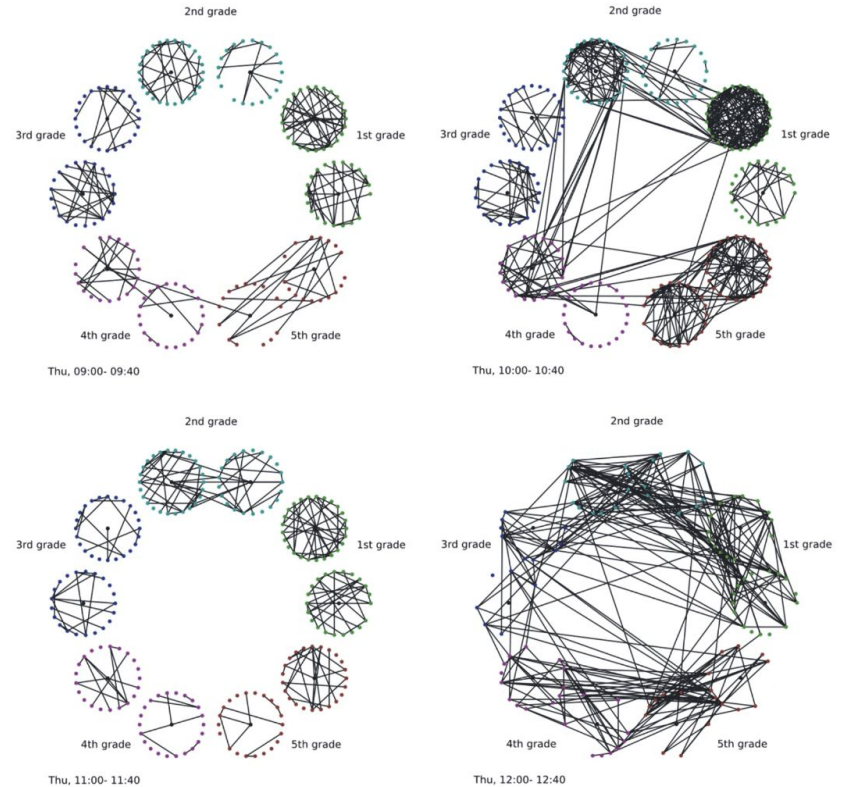


# Temporal Networks and the Activity Driven Framework

Nicolò Gozzi, ISI Foundation  
October 25, 2023

# What is a Temporal Network?

A temporal (or time-varying) network is a network whose links, or nodes, are **active only at certain times**



# Examples of Temporal Networks

- Person-to-person communication (emails, text messages, phone calls)
- Physical proximity (face-to-face interactions)
- Cell biology (protein interaction, gene-regulatory networks)
- Infrastructural networks (air-transport network)
- Neural and brain networks
- Ecological networks (seasonality, environmental changes)
- ...

# Examples of Temporal Networks

- Person-to-person communication (emails, text messages, phone calls)
  - Physical proximity (face-to-face interactions)
  - Cell biology (protein interaction, gene-regulatory networks)
  - Infrastructural networks (air-transport network)
  - Neural and brain networks
  - Ecological networks (seasonality, environmental changes)
  - ...
- The study of temporal network is profoundly related to the **study of dynamical phenomena** unfolding on such networks (e.g., epidemics)

# **It's all about Timescales**

Whether to represent or not a network as a time-varying network depends on the time scale of process we are studying:

# It's all about Timescales

Whether to represent or not a network as a time-varying network depends on the time scale of process we are studying:

- When  $t_N \gg t_P \rightarrow$  **Static Approximation**: the timescale of network evolution is much greater than the timescale of dynamical process on the network. It is safe to use a static approximation of the network

# It's all about Timescales

Whether to represent or not a network as a time-varying network depends on the time scale of process we are studying:

- When  $t_N \gg t_P \rightarrow$  **Static Approximation**: the timescale of network evolution is much greater than the timescale of dynamical process on the network. It is safe to use a static approximation of the network
- When  $t_N \sim t_P \rightarrow$  **Temporal Network**: the timescale of network evolution and of the dynamical process is comparable. It is convenient to use a time-varying network representation

# It's all about Timescales

Whether to represent or not a network as a time-varying network depends on the time scale of process we are studying:

- When  $t_N \gg t_P \rightarrow$  **Static Approximation**: the timescale of network evolution is much greater than the timescale of dynamical process on the network. It is safe to use a static approximation of the network
- When  $t_N \sim t_P \rightarrow$  **Temporal Network**: the timescale of network evolution and of the dynamical process is comparable. It is convenient to use a time-varying network representation
- When  $t_N \ll t_P \rightarrow$  **Annealed Approximation**: the timescale of network evolution is much faster than the dynamical processes. It is convenient to use a time-aggregated (static) version of the network

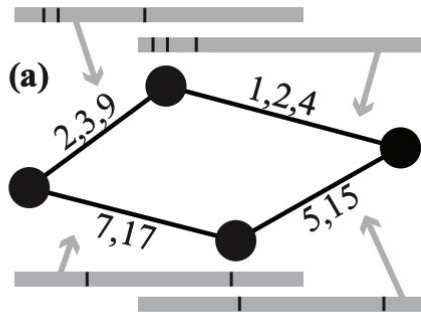


# Representation of Temporal Networks: Contact Sequences

Depending on their features, temporal networks can be represented as:

## 1. Contact Sequences:

- If **duration of interactions is negligible**, the network can be represented as a contact sequence  $C$  made of triples  $(i, j, t)$  where  $i, j \in V$  and  $t$  denotes time.
- Examples include communication data, and physical proximity data where the duration of the contact is negligible.
- Same time contacts are grouped into one graph the temporal network is presented as a **sequence of static graphs**



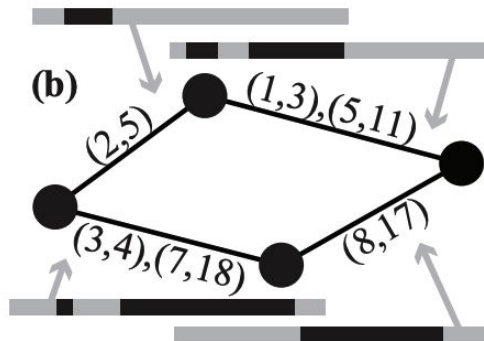
# Representation of Temporal Networks: Interval Graphs

Depending on their features, temporal networks can be represented as:

## 2. Interval Graphs:

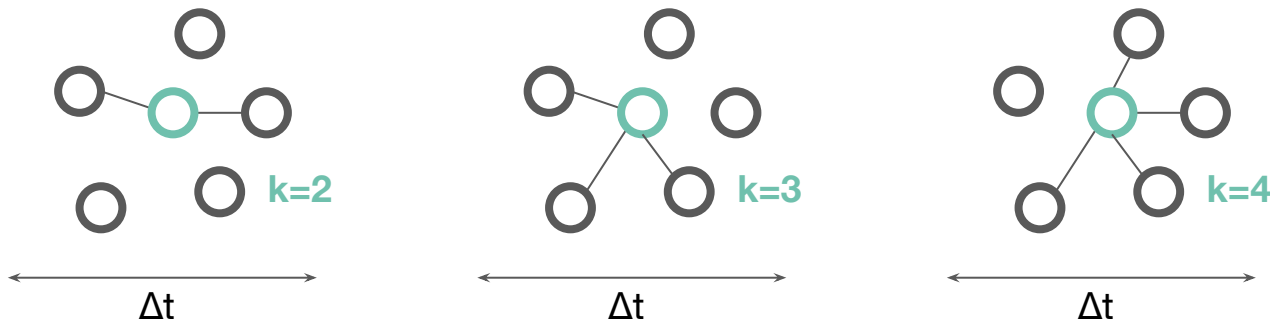
- in this case, the edges are active over a set of intervals time intervals  $T_e = \{(t_1, t'_1), \dots (t_n, t'_n)\}$
- Examples include proximity contact networks
- Similarly to static graph, it can be defined an adjacency index:

$$a(i, j, t) = \begin{cases} 1 & \text{if } i \text{ and } j \text{ are connected at time } t \\ 0 & \text{otherwise} \end{cases}$$



# Topological Measures for Temporal Network

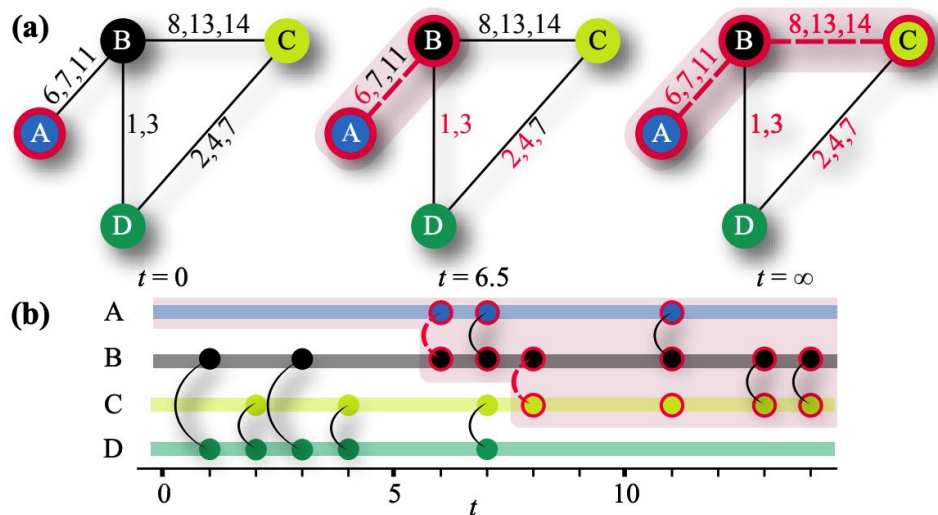
- Many of the metrics developed for static networks can still be applied to temporal networks
- These measures can be computed on the **aggregated network** over a certain time interval or over regular time steps
- In doing so, we can compute, for example, the **time-dependent degree** of a node



- However, other properties are directly influenced by the order of link activations

# Time-respecting paths and reachability

- In a static graph, a path is simply a sequence of sequential edges
- In temporal networks, **paths are sequences of link ordered in time**



# Time-respecting paths and reachability

- In a static graph, a path is simply a sequence of sequential edges
- In temporal networks, **paths are sequences of link ordered in time**
- More formally, **time-respecting paths** are defined as sequences of contacts with **non-decreasing times**
- The set of vertices that can be reached by time-respecting paths from node  $i$  within an observation window is called the **set of influence** of  $i$
- The average fraction of nodes in the sets of influence of all vertices as the **reachability ratio**

# Connectivity and components

- Connectivity measures if a **pair of nodes is connected by a path**
- Connectivity is **not a symmetric relation for temporal graphs** ( $i$  may be reachable by time-respecting paths from  $j$ , but not the opposite)
- We can define:
  - two nodes  $i$  and  $j$  of a temporal network are **strongly connected** if there is a directed, time-respecting path connecting  $i$  to  $j$
  - two nodes  $i$  and  $j$  are **weakly connected** if there are undirected time respecting paths from  $i$  to  $j$  and  $j$  to  $i$  (i.e. the directions of the contacts are not taken into account)
- Based on this, we can define **strongly or weakly connected components** of the temporal graph

# Distances, Latencies, and Fastest Paths

- For static networks, the **distance between two nodes** is defined as the length of the shortest path joining them (in terms of n. of edges)
- For temporal network, one can consider the fastest time-respecting path(s) between two nodes
- The shortest time within which  $i$  can reach  $j$  is called their **latency** (also temporal distance)

# **The Activity Driven Framework**



# The Activity Driven Framework

Classic Network Growth Models (Erdős-Rényi, preferential attachment model) are **connectivity-driven models**:

- Network's topology is at the core of the models' definition
- Well-suited to capture the features of network systems whose links among nodes are long-lived elements
- However, in some systems interactions may rapidly change on a very short time scale

The activity-driven model is a **temporal network model** that can be used to describe those systems

# The Activity Driven Framework

The core assumption of the activity driven model is that each node  $i$  has an **activity potential**  $a_i$ , defined as the **probability per unit time to create new interactions**

## Network Initialization:

- Consider  $N$  disconnected nodes
- Each node is assigned an activity sampled from a distribution  $F(a)$
- To avoid divergences of  $F(a)$  close to the origin we assume  $\varepsilon \leq a \leq 1$

# The Activity Driven Framework

Once activity has been assigned to each node, the temporal dynamics of the model **at each time step** can be defined as follows:

# The Activity Driven Framework

Once activity has been assigned to each node, the temporal dynamics of the model **at each time step** can be defined as follows:

1. At each discrete time step  $t$  the network  $G_t$  starts with  $N$  disconnected nodes

# The Activity Driven Framework

Once activity has been assigned to each node, the temporal dynamics of the model **at each time step** can be defined as follows:

1. At each discrete time step  $t$  the network  $G_t$  starts with  $N$  disconnected nodes
2. **Node Activation:** With probability  $a_i \Delta t$  each node  $i$  becomes active

# The Activity Driven Framework

Once activity has been assigned to each node, the temporal dynamics of the model **at each time step** can be defined as follows:

1. At each discrete time step  $t$  the network  $G_t$  starts with  $N$  disconnected nodes
2. **Node Activation:** With probability  $a\Delta t$  each node  $i$  becomes active
3. **Partner Selection:** each active node generates  $m$  links that are connected to  $m$  other randomly selected nodes (non-active nodes can still receive connections from other active vertices)

# The Activity Driven Framework

Once activity has been assigned to each node, the temporal dynamics of the model **at each time step** can be defined as follows:

1. At each discrete time step  $t$  the network  $G_t$  starts with  $N$  disconnected nodes
2. **Node Activation:** With probability  $a\Delta t$  each node  $i$  becomes active
3. **Partner Selection:** each active node generates  $m$  links that are connected to  $m$  other randomly selected nodes (non-active nodes can still receive connections from other active vertices)
4. At the next time step  $t + \Delta t$ , all the edges in the network  $G_t$  are deleted (underlying assumption is that all interactions have a constant duration  $\tau_i = \Delta t$ )

# The Activity Driven Framework

Once activity has been assigned to each node, the temporal dynamics of the model **at each time step** can be defined as follows:

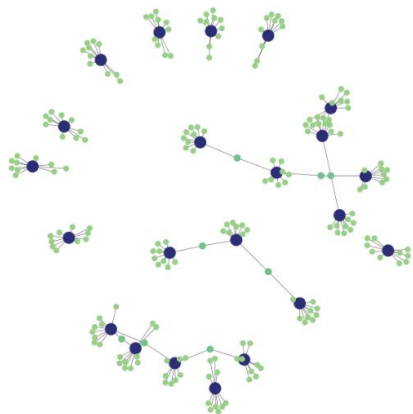
1. At each discrete time step  $t$  the network  $G_t$  starts with  $N$  disconnected nodes
2. **Node Activation:** With probability  $a\Delta t$  each node  $i$  becomes active
3. **Partner Selection:** each active node generates  $m$  links that are connected to  $m$  other randomly selected nodes (non-active nodes can still receive connections from other active vertices)
4. At the next time step  $t + \Delta t$ , all the edges in the network  $G_t$  are deleted (underlying assumption is that all interactions have a constant duration  $\tau_i = \Delta t$ )



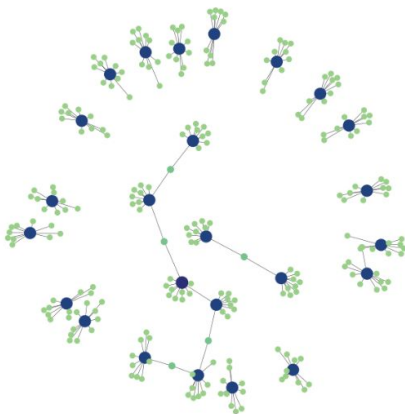
# ADN - Example

We consider  $F(a) \sim a^{-\gamma}$ , with  $\gamma = 2.1$  and  $m = 10$

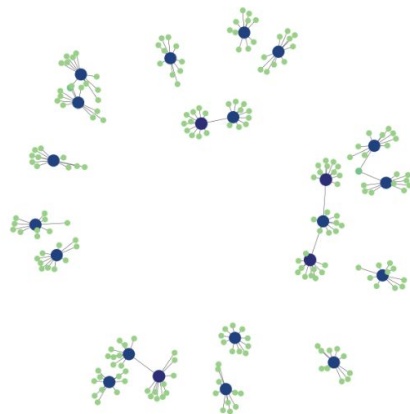
T=1



T=10



T=20



At each time step, the instantaneous network will be composed by a **set of stars around the nodes that were active** at that time step

# ADN - Time-aggregated Network

We define the integrated network as the union of all the networks obtained in each previous time step:

$$G_T = \bigcup_{t=0}^{t=T} G_t$$

It can be shown that node  $i$  will have at time  $T$  a degree in the integrated network given by:

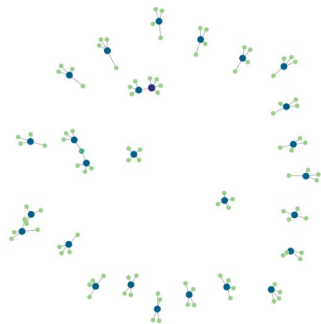
$$k_i(T) = N(1 - e^{-Tm\mu x_i/N})$$

And the degree distribution  $P_T(k)$  of the integrated network at time  $T$  is:

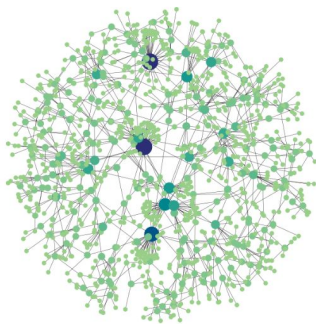
$$P_T(k) \sim F[k / (T \cdot m)]$$

# ADN - Time-aggregated Network

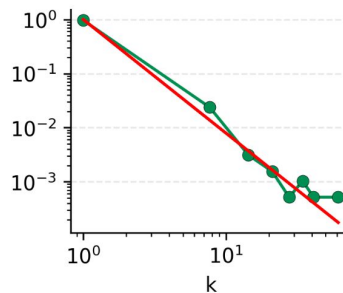
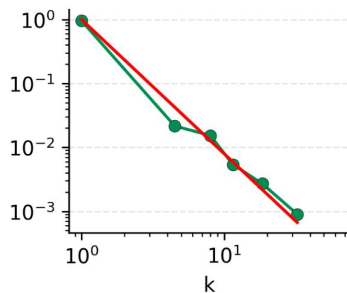
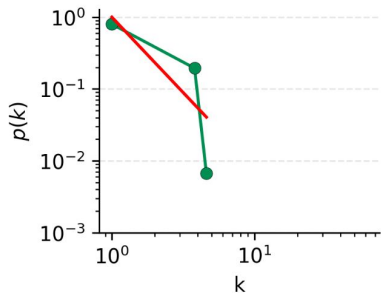
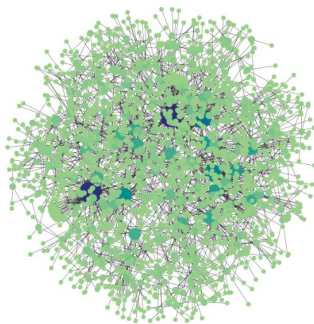
T=1



T=10



T=20



The **degree distribution** in the time-aggregated network follows the **distribution of the activity**

Hence, heterogeneous activity patterns induce the **formation of hubs** (i.e., highly active nodes engaging over and over in social interactions)

# ADN - Limitation

# ADN - Limitation

The presented model is **Markovian**: nodes do not have memory of the previous time steps.

# ADN - Limitation

The presented model is **Markovian**: nodes do not have memory of the previous time steps.

- **Pros**: The full dynamics of the network and its ensuing structure is thus completely encoded in the activity potential distribution  $F(x)$

# ADN - Limitation

The presented model is **Markovian**: nodes do not have memory of the previous time steps.

- **Pros**: The full dynamics of the network and its ensuing structure is thus completely encoded in the activity potential distribution  $F(x)$
- **Cons**: since links are created at random, the distribution of **links' weights** in the time-integrated network **is homogeneous** and thus very far from observations in real networks

# ADN - Extension to Communities

- Real complex networks are organised into tightly connected groups → **communities** (e.g., circles of friends)
- As a result, the vast majority of **connections takes place within communities** rather than across them
- The ADN framework can be extended to go **beyond random partner selection** and include also communities



# ADN - Extension to Communities

Network Initialization

# ADN - Extension to Communities

## Network Initialization

- A. Each node is assigned an activity  $a_i$  sampled from  $F(a)$

# ADN - Extension to Communities

## Network Initialization

- A. Each node is assigned an activity  $a_i$  sampled from  $F(a)$
- B. Each node is assigned to a particular community  $c$ . The size  $s$  of each community is extracted from a distribution  $G(s)$

# ADN - Extension to Communities

## Network Initialization

- A. Each node is assigned an activity  $a_i$  sampled from  $F(a)$
- B. Each node is assigned to a particular community  $c$ . The size  $s$  of each community is extracted from a distribution  $G(s)$

## Network Dynamics

# ADN - Extension to Communities

## Network Initialization

- A. Each node is assigned an activity  $a_i$  sampled from  $F(a)$
- B. Each node is assigned to a particular community  $c$ . The size  $s$  of each community is extracted from a distribution  $G(s)$

## Network Dynamics

1. At each time  $t$ , the network  $G_t$  is initially disconnected

# ADN - Extension to Communities

## Network Initialization

- A. Each node is assigned an activity  $a_i$  sampled from  $F(a)$
- B. Each node is assigned to a particular community  $c$ . The size  $s$  of each community is extracted from a distribution  $G(s)$

## Network Dynamics

- 1. At each time  $t$ , the network  $G_t$  is initially disconnected
- 2. Each node is active with probability  $a\Delta t$  creating  $m$  connections

# ADN - Extension to Communities

## Network Initialization

- A. Each node is assigned an activity  $a_i$  sampled from  $F(a)$
- B. Each node is assigned to a particular community  $c$ . The size  $s$  of each community is extracted from a distribution  $G(s)$

## Network Dynamics

1. At each time  $t$ , the network  $G_t$  is initially disconnected
2. Each node is active with probability  $a\Delta t$  creating  $m$  connections
3. With probability  $\eta$  each connection is done selecting at random one of the nodes in the same community and with probability  $1 - \eta$  selecting at random in any other community

# ADN - Extension to Communities

## Network Initialization

- A. Each node is assigned an activity  $a_i$  sampled from  $F(a)$
- B. Each node is assigned to a particular community  $c$ . The size  $s$  of each community is extracted from a distribution  $G(s)$

## Network Dynamics

1. At each time  $t$ , the network  $G_t$  is initially disconnected
2. Each node is active with probability  $a\Delta t$  creating  $m$  connections
3. With probability  $\eta$  each connection is done selecting at random one of the nodes in the same community and with probability  $1 - \eta$  selecting at random in any other community
4. Each connection is deleted, time incremented to  $t + \Delta t$  and the process restarts from the first point



# ADN - Extension to Communities

## Network Initialization

- A. Each node is assigned an activity  $a_i$  sampled from  $F(a)$
- B. Each node is assigned to a particular community  $c$ . The size  $s$  of each community is extracted from a distribution  $G(s)$

## Network Dynamics

1. At each time  $t$ , the network  $G_t$  is initially disconnected
2. Each node is active with probability  $a\Delta t$  creating  $m$  connections
3. With probability  $\eta$  each connection is done selecting at random one of the nodes in the same community and with probability  $1 - \eta$  selecting at random in any other community
4. Each connection is deleted, time incremented to  $t + \Delta t$  and the process restarts from the first point

# ADN - Extension to Communities

The parameter  $\eta$  **regulates the modularity** of the emerging network:

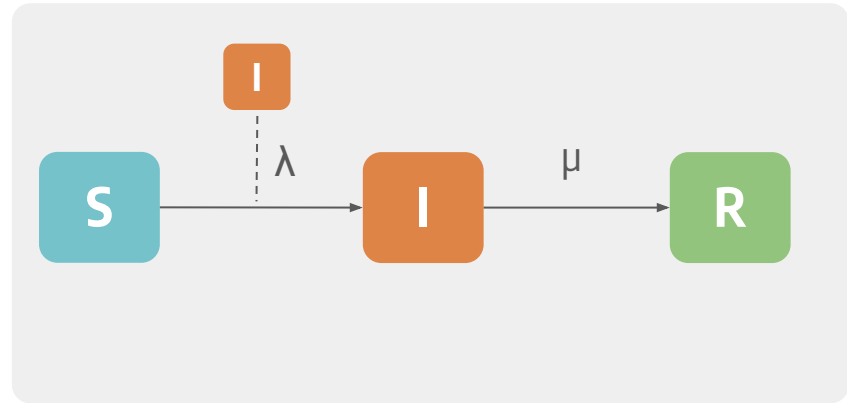
- For  $\eta = 0$  (and in case of community sizes  $s \ll N$ ) the network unfolds very similarly to the random partner selection case
- Instead for  $\eta = 1$  the network will be formed by completely disconnected communities.

# **Epidemic Spreading on ADN**

# Epidemic Spreading on ADN - SIR Model

SIR compartmental model:

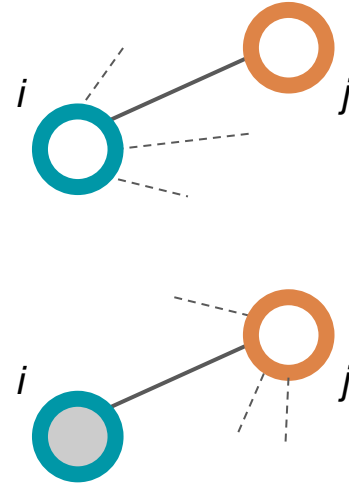
- Nodes are divided into 3 groups: susceptible, infected, and recovered
- By interacting with the infected, susceptibles may get infected at rate  $\lambda$
- Infected nodes recover spontaneously at rate  $\mu$



# Epidemic Spreading on ADN - SIR Model

Nodes can become infected in two ways at each time step:

1. Node  $i$  is **active**, establish a connection with an infected node  $j$  and acquire the infection from  $j$
2. Node  $i$  is **inactive**, but is it contacted by infected and active node  $j$  that infects  $i$



## Epidemic Spreading on ADN - SIR Model (Equations)

$$d_t S_a = -m\lambda S_a a \int da' \frac{I_{a'}}{N} - m\lambda \int da' \frac{I_{a'} a'}{N}$$

$$d_t I_a = -\mu I_a + m\lambda S_a a \int da' \frac{I_{a'}}{N} + m\lambda \int da' \frac{I_{a'} a'}{N}$$

$$d_t R_a = \mu I_a$$

# Epidemic Spreading on ADN - SIR Model (Equations)

SIR dynamics on ADN, can be written down as the following system of differential equations:

$$d_t S_a = -m\lambda S_a a \int da' \frac{I_{a'}}{N} - m\lambda S_a \int da' \frac{I_{a'} a'}{N}$$

$$d_t I_a = -\mu I_a + m\lambda S_a a \int da' \frac{I_{a'}}{N} + m\lambda S_a \int da' \frac{I_{a'} a'}{N}$$

$$d_t R_a = \mu I_a$$

# Epidemic Spreading on ADN - SIR Model (Equations)

SIR dynamics on ADN, can be written down as the following system of differential equations:

$$d_t S_a = -m\lambda S_a a \int da' \frac{I_{a'}}{N} - m\lambda S_a \int da' \frac{I_{a'} a'}{N}$$

$$d_t I_a = -\mu I_a + m\lambda S_a a \int da' \frac{I_{a'}}{N} + m\lambda S_a \int da' \frac{I_{a'} a'}{N}$$

$$d_t R_a = \mu I_a$$



# Epidemic Spreading on ADN - SIR Model (Equations)

SIR dynamics on ADN, can be written down as the following system of differential equations:

$$d_t S_a = \underbrace{-m\lambda S_a a \int da' \frac{I_{a'}}{N}}_{\text{infection rate}} - m\lambda S_a \int da' \frac{I_{a'} a'}{N}$$

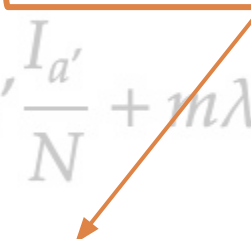
$$d_t I_a = -\mu I_a + m\lambda S_a a \int da' \frac{I_{a'}}{N} + m\lambda S_a \int da' \frac{I_{a'} a'}{N}$$

$$d_t R_a = \mu I_a$$

Number of susceptible nodes that are active and become infected contacting an infected node

# Epidemic Spreading on ADN - SIR Model (Equations)

SIR dynamics on ADN, can be written down as the following system of differential equations:

$$d_t S_a = -m\lambda S_a a \int da' \frac{I_{a'}}{N} - m\lambda S_a \int da' \frac{I_{a'} a'}{N}$$
$$d_t I_a = -\mu I_a + m\lambda S_a a \int da' \frac{I_{a'}}{N} + m\lambda S_a \int da' \frac{I_{a'} a'}{N}$$


$$d_t R_a = \mu I_a$$

Number of susceptible nodes that are inactive and become infected when contacted by an active infected node

# Epidemic Spreading on ADN - SIR Model (Equations)

SIR dynamics on ADN, can be written down as the following system of differential equations:

$$d_t S_a = -m\lambda S_a a \int da' \frac{I_{a'}}{N} - m\lambda S_a \int da' \frac{I_{a'} a'}{N}$$

$$d_t I_a = \underbrace{-\mu I_a}_{\text{Number of infected nodes that spontaneously recover at rate } \mu} + m\lambda S_a a \int da' \frac{I_{a'}}{N} + m\lambda S_a \int da' \frac{I_{a'} a'}{N}$$

$$d_t R_a = \mu I_a$$

Number of infected nodes that spontaneously recover at rate  $\mu$

# SIR Model on ADN - Epidemic Threshold

Using a mean-field approach, we can derive the epidemic threshold of the SIR model on memory-less activity driven networks:

$$\frac{\lambda}{\mu} > \frac{1}{m} \frac{1}{\langle a \rangle + \sqrt{\langle a^2 \rangle}}$$

And it follows that the Basic Reproduction Number is:

$$R_0 = m \frac{\lambda}{\mu} \left( \langle a \rangle + \sqrt{\langle a^2 \rangle} \right)$$

- If  $R_0 > 1$  the epidemic will be able to spread
- if  $R_0 < 1$  the initial outbreak will not result in a macroscopic epidemic

# References

- Holme, P., & Saramäki, J. (2012). Temporal networks. *Physics reports*, 519(3), 97-125.
- Masuda, N., & Lambiotte, R. (2016). *A guide to temporal networks*.
- Holme, P., & Saramäki, J. (Eds.). (2019). *Temporal network theory* (Vol. 2). New York: Springer.
- Sociopatterns: <http://www.sociopatterns.org/>
- Perra, N., Gonçalves, B., Pastor-Satorras, R., & Vespignani, A. (2012). Activity driven modeling of time varying networks. *Scientific reports*, 2(1), 469.
- Liu, S., Perra, N., Karsai, M., & Vespignani, A. (2014). Controlling contagion processes in activity driven networks. *Physical review letters*, 112(11), 118702.

**Extra**

# SIR Model on ADN - Epidemic Threshold

We consider the evolution of the number of infected node of activity class  $a$ :

$$d_t I_a = -\mu I_a + m\lambda S_a a \int da' \frac{I_{a'}}{N} + m\lambda S_a \int da' \frac{I_{a'} a'}{N}$$

Since  $N_a = S_a + I_a + R_a$  we can substitute  $S_a = N_a - I_a - R_a$ :

$$d_t I_a = -\mu I_a + m\lambda(N_a - I_a - R_a)a \int da' \frac{I_{a'}}{N} + m\lambda(N_a - I_a - R_a) \int da' \frac{I_{a'} a'}{N}$$

# SIR Model on ADN - Epidemic Threshold

In early stage assumption we have that  $R_a \sim 0$  and  $I_a \sim 0$ . Therefore, we drop all the second order terms obtaining:

$$d_t I_a = -\mu I_a + m\lambda N_a a \int da' \frac{I_{a'}}{N} + m\lambda N_a \int da' \frac{I_{a'} a'}{N}$$

By integrating both sides over all activity classes we have:

$$d_t I = -\mu I + m\lambda \langle a \rangle I + m\lambda \Theta$$

where  $\Theta = \int da I_a a$  and  $\langle a^n \rangle = \int da F(a) a^n$



# SIR Model on ADN - Epidemic Threshold

To characterise the evolution of  $I(t)$  we then need to derive an equation for  $\Theta$ . In particular, multiplying both sides of previous equation by  $a$  and integrating across all activities we get:

$$d_t \Theta = -\mu \Theta + m\lambda \langle a^2 \rangle I + m\lambda \langle a \rangle \Theta$$

The epidemic threshold can be obtained studying the stability of the system of differential equations defined by:

$$d_t I = -\mu I + m\lambda \langle a \rangle I + m\lambda \Theta$$

$$d_t \Theta = -\mu \Theta + m\lambda \langle a^2 \rangle I + m\lambda \langle a \rangle \Theta$$

# SIR Model on ADN - Epidemic Threshold

Indeed, the disease will be able to spread only if the largest eigenvalue of the Jacobian matrix  $J$  of the system is larger than zero. Here,  $J$  can be written as:

$$\begin{pmatrix} -\mu + m\lambda\langle a \rangle & m\lambda \\ m\lambda\langle a^2 \rangle & -\mu + m\lambda\langle a \rangle \end{pmatrix}$$

By solving the eigenvalue problem for this matrix, and imposing positivity of largest eigenvalue, we get the threshold:

$$\frac{\lambda}{\mu} > \frac{1}{m} \frac{1}{\langle a \rangle + \sqrt{\langle a^2 \rangle}}$$

And therefore the  $R_0$  reads:

$$R_0 = m \frac{\lambda}{\mu} \left( \langle a \rangle + \sqrt{\langle a^2 \rangle} \right)$$