

Statistical Network Analysis

Prof. Dr. Ingo Scholtes

Chair of Machine Learning for Complex Networks
Center for Artificial Intelligence and Data Science (CAIDAS)

Julius-Maximilians-Universität Würzburg
Würzburg, Germany

ingo.scholtes@uni-wuerzburg.de

Lecture 10
Random Walks and Markov Chains

January 11, 2023



Notes:

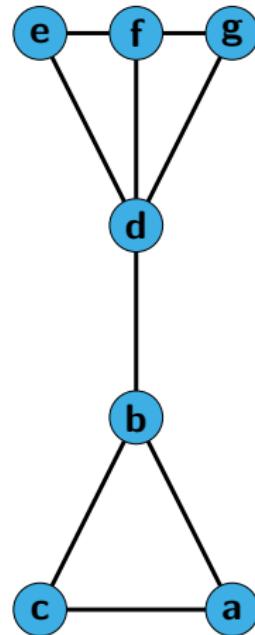
- **Lecture L10:** Random Walks and Markov Chains 11.01.2023
- **Educational Objective:** Moving from the analysis of the structure to dynamics on networks, we introduce Markov chain models for dynamical processes on networks. We cover random walk models, which are an important foundation for network analysis and graph learning.
 1. Dynamical processes on complex networks
 2. Random Walks and Markov Chains
 3. Markov chain convergence theorem
- **Exercise 08:** Random Walks and Diffusion Speed due 18.01.2023

From structure to dynamics ...

- ▶ we studied the **topology of networks**
 - ▶ community structure and clustering
 - ▶ node centralities
 - ▶ degree-based ensembles
 - ▶ connectivity, diameter, robustness
- ▶ networks are “substrate” or “infrastructure” for **dynamical processes**
- ▶ interplay b/w structure and dynamics crucial to understand **function of networked systems**

open questions

- ▶ which types of dynamical processes exist?
- ▶ how can we model such processes in networks?
- ▶ what can we learn about the network?
- ▶ what can we learn about nodes?



Notes:

- In the last lecture of 2021, we concluded the chapter on stochastic models of complex networks (cf. course outline in lecture L01). In this chapter we studied ensemble-based methods to derive statements about expected properties of **random networks** under a given set of constraints (e.g. their size, number of links or degree sequence/distribution).
- We concluded this chapter of the course with some **cautionary** statements on the use of ensemble-based method for the study of real-world networked systems.
- All lectures so far focused on the **topology** of a network, while ignoring the **function** of the network. But networks are typically a “substrate” or “infrastructure” (note the etymology of these Latin terms) for dynamical processes running **on** the topology.
- Examples for dynamical processes include the **propagation** of information, rumours or failures, the **synchronization** of some node dynamics (e.g. in neural networks or oscillator networks), or the diffusion of physical quantities.
- The topology of a network naturally influences how such processes evolve. Hence, if we want to understand the function of a system we need methods to relate the network topology with the evolution of dynamical processes.

Dynamical processes on networks

dynamical processes in networks

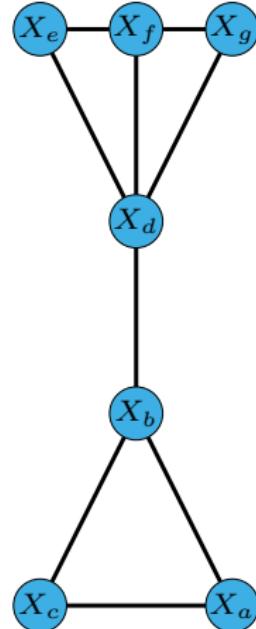
for node v with neighbors n_1, \dots, n_k consider a **dynamical system**

$$\dot{x}_v = f(x_{n_1}, x_{n_2}, \dots, x_{n_k}) \text{ where}$$

- ▶ \dot{x}_v is **Newton's notation** for time derivative of **time-evolving state** $x_v(t)$ of node v
 - ▶ f is a linear or non-linear **function**
 - ▶ t is discrete or continuous **time**
-
- ▶ we assume that states of nodes evolve while the **network topology is static** → L13 - Temporal Networks

important classes of dynamical processes

- ▶ synchronization
- ▶ consensus
- ▶ propagation
- ▶ diffusion



Notes:

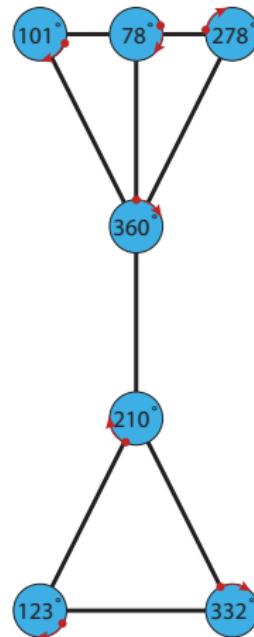
- We first clarify what exactly we want to model. We consider a **static network topology** as well as **time-evolving states of nodes**. For the moment, the assumption that the topology is static is important but we will later see that the techniques introduced in today's lecture can be generalized to model dynamical processes in dynamic networks.
- We further assume that the evolution of states is described by a **dynamical system** which can capture, e.g., spreading or propagation processes, diffusion, consensus dynamics or synchronization. In general these dynamical systems can be linear or non-linear and we will consider both discrete and continuous time.
- Typical processes studied in the context of network analysis can be broadly categorised in four different classes. Note that (unfortunately), this distinction is often not made very clear and researchers often – in my opinion wrongly– refer to consensus models as “diffusion” or “spreading” processes.
- Today we will consider a particularly simple case: a discrete time linear dynamical system capturing a diffusion process. In the next lecture, we generalise this to continuous time, which will allow us to highlight interesting relations between the evolution of dynamical systems in networks vs. Euclidean space.

Synchronization in networks

- ▶ we can study **networks of coupled oscillators**
 - ▶ nodes generate oscillating signal
 - ▶ oscillations of nodes influence oscillation frequencies of connected nodes
- ▶ example models include **Kuramoto model**, **Mirollo-Strogatz model**, **Winfree model**
- ▶ mostly **continuous-time** and **non-linear** processes

exemplary systems and phenomena

- ▶ neural oscillations/spikes, brainwaves
- ▶ cardiac pacemaker cells
- ▶ firefly synchronization
- ▶ synchronization of protocol messages
- ▶ frequency synchronization in AC grids



Notes:

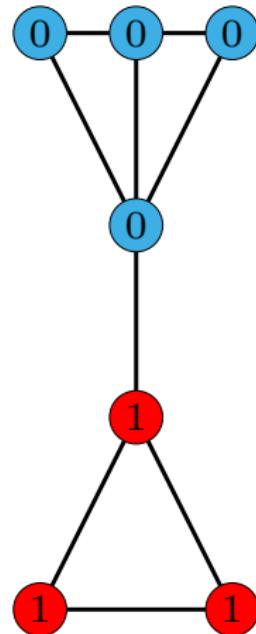
- But let us first motivate different classes of processes by studying some simple examples. The first class of models captures synchronization phenomena in complex networks. Here we often assume continuous, periodic states (e.g., $x_i \in [0, 2\pi]$) as well as continuous time. Different from models where nodes are “passive” in the absence of an external “force”, here we often consider models where nodes exhibit an **intrinsic dynamics**.
- For instance, each node can be an **oscillator** that – by itself – **oscillates** at an intrinsic frequency. We additionally consider interactions between connected nodes, e.g. depending on the “angles” between periodic signals, nodes advance or reduce their internal **oscillation** frequency.
- In the resulting dynamical systems, interesting questions about the emergence of collective dynamics (such as globally synchronised states) emerge. This is related to complex biological functions like, e.g. the emergence of synchronised cell activities which is the basis for the cardiac pacemaker, circadian patterns, or neuronal activities. A nice review of network-based studies of such synchronisation phenomena can be found in → A Arenas et al., 2008
- A very nice popular science account of the study of synchronisation in science is given in the book “Sync” by Steven Strogatz.

Consensus in networks

- ▶ we can study **opinion and consensus formation**
 - ▶ nodes have discrete or continuous “opinions”
 - ▶ nodes influence/adopt opinion of neighbors over time
- ▶ can lead to complex **collective behaviour**
- ▶ exemplary models include **Ising model, Voter model, Vicsek model, Deffuant-Weisbuch model, Axelrod model**

exemplary systems and phenomena

- ▶ opinion formation in societies
- ▶ cultural dynamics
- ▶ behaviour in stock markets
- ▶ ferromagnetism
- ▶ swarming and flocking behavior



Notes:

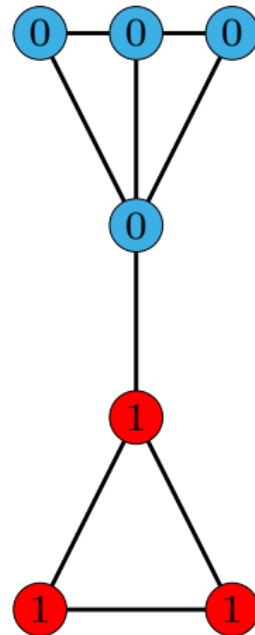
- Under certain conditions, synchronisation models can produce a particular type of **consensus among nodes** (where consensus refers to a single common oscillatory pattern across all nodes).
- But we can also consider simpler models for consensus formation in networks. Consider, for instance, nodes with discrete states (as well as a discrete notion of time). Based on an initial distribution of states we can consider a dynamical system where in each step of the process each node updates its state based on the state of its neighbours. We can, e.g., simply define a rule that each node adopts the state of the majority of its neighbours. Such models are the simplest possible “null models” for opinion formation in networks, which is why they are sometimes called “opinion dynamics models”.
- Clearly, this is a very simple and unrealistic model for “opinions” of people in a real social network. To get meaningful and interesting results, we thus have to incorporate further ingredients like, e.g. heterogeneous properties such as the “williness” of a node to be persuaded by its neighbours, different socio-economic backgrounds that may affect opinions, asymmetric access to information, etc.

Propagation in networks

- ▶ we can study how non-conserved quantities **propagate** through networks
- ▶ mostly discrete-time processes
- ▶ exemplary models include
susceptible/infected (SI) model, threshold models, Kleinberg cascade model

exemplary systems and phenomena

- ▶ disease spreading
- ▶ propagation of rumours/fake news in social networks
- ▶ information cascades
- ▶ failure cascades in power grids



Notes:

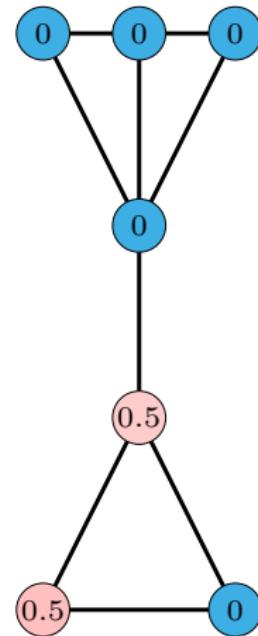
- A next class of widely studied models captures **propagation** processes. The Latin word “propagare” means: to extend, to increase, to multiply. This emphasizes that we consider a process by which some initial quantities (or properties) **multiply or proliferate** through the network.
- Real-world examples for such processes include the infection of nodes with diseases, the **proliferation** of rumours, the sharing of information or the propagation of failures (e.g. the kind of dynamical process that we have explicitly excluded in our study of robustness in lecture L07).
- In all of these cases we have a **non-conserved quantity**, i.e. the fact that you infect your friend with the flu, unfortunately does not reduce your own infection. Moreover, having infected your friend does not affect your ability to infect others. The same holds for rumours, failures or information which can freely multiply in a network.

Diffusion in networks

- ▶ we can model **diffusion of conserved quantity** in a network
 - ▶ each node contains a **quantity of something**
 - ▶ quantities can move/flow between nodes
 - ▶ total quantity is conserved at all times
- ▶ continuous- or discrete-time models

two interpretations

1. **divisible quantity** of water, cars, money, etc. at given node
2. **probability** that **indivisible object** is at given node (implies that total “quantity” always sums to one)



- ▶ in this chapter, we focus on **statistical models of diffusion in networks**

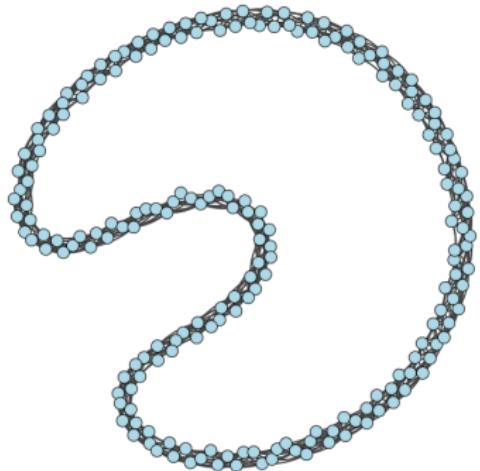
Notes:

- This property of propagation processes is in contrast to **diffusion processes**. Note that the latin word “diffundere” means to “spread out” or to “pour out”.
- Here we consider an initial quantity that is **conserved** and that **does not** multiply throughout the process. We rather distribute it through the network similar to how we spread butter on a toast. Even though diffusion and propagation processes are related, we should not treat them as the same.
- In the example above, one unit of a (divisible) quantity in the top left node is spread out among the two neighbours, in this example each neighbour receiving half of the quantity (we can also consider other ways to distribute it).
- Important examples for diffusion processes include the diffusion of conserved (physical) quantities like water, gas, electricity, money, cars, people, etc. Here, the numbers in the example above indicate the share of the quantity that a node receives over time.
- However, what about cases where a single person, car, or atom “diffuses” through the network, i.e. where we have an “indivisible” object that changes its location. We can still think of such scenarios as a diffusion process, but we now interpret the “quantity” that diffuses as the “probability” of the indivisible object to be at a given point in time at a given node.

Example: diffusion in networks

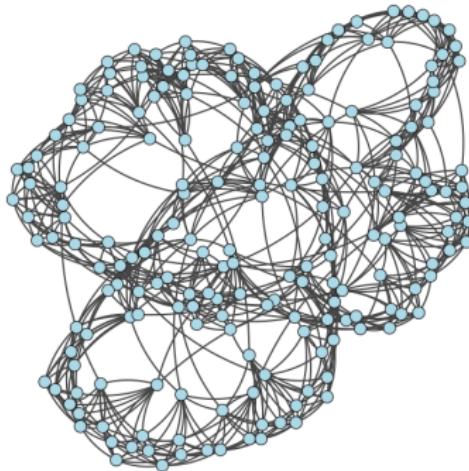
example 1

simulation of diffusion process in one-dimensional Watts-Strogatz network with $s = 5$ and $p = 0$



example 2

simulation of diffusion process in one-dimensional Watts-Strogatz network with $s = 5$ and $p = 0.1$



questions to answer

- ▶ how does the network topology influence diffusion speed?
- ▶ what can we learn about the importance of nodes?

Notes:

- In today's and next week's lecture, we will study diffusion processes in more detail.
- The examples above show the evolution of such a diffusion process in two networks. Specifically, I have used the Watts-Strogatz model to generate a regular ring lattice (left) and a small-world network with small diameter and large clustering coefficient (right). Both networks have the same number of nodes and links but the two networks have different topologies that influence the diffusion process.
- In the example above, the topology has huge implications for the evolution of the process. In particular, due to the large diameter and the bottlenecks in the network the diffusion process is slower than in the example on the right. But how can we understand and predict this effect on diffusion? For this we need to mathematically model and analyse diffusion processes.

Modeling diffusion

example from practice

- ▶ you loose your drunk friend in the streets of Würzburg
- ▶ you need to search for him/her?

- ▶ we model the streets of Würzburg as network
- ▶ you do not know which walk your friend might have taken in this network
- ▶ we can model **lack of knowledge** about actual path by assigning a **probability** to each possible walk
- ▶ we use statistical model to analyse **dynamical process** on a network



Map of Würzburg

image credit: Map created from openstreetmap data

Notes:

- Before diving into the mathematical details of diffusion models, we introduce a practical example in which you are required to **model a process on a network** whose details you – unfortunately – don't know.
- We consider a situation where you have lost your drunk friend in the streets of Würzburg. All you know is where you have last seen your friend. But when you arrive there, he or she is gone. Where should you search to maximise the probability to find your friend?
- An understanding of processes in networks can help us to solve this problem. In particular, we can see this as an example for a “diffusion” process in a network. Your friend is (hopefully) a “conserved quantity” that walks the network of streets in Würzburg.
- Your task is to predict the current location of your friend given that you know (i) the network topology, (ii) the initial position of our friend, and (iii) how much time has passed since you lost her or him.
- Since we do not know the specific path that our friend has taken, we need a model that captures our lack of knowledge about “the process” (i.e. how our friend walks through the streets of Würzburg).
- So, much like we used probabilities to formalise our lack of knowledge about microstates in the study of statistical ensembles, we now use stochastic models to formalise our “lack of knowledge” about the evolution of the “process”. We can, for instance, use such a stochastic model to calculate the probabilities of all paths that your friend **could** have taken.

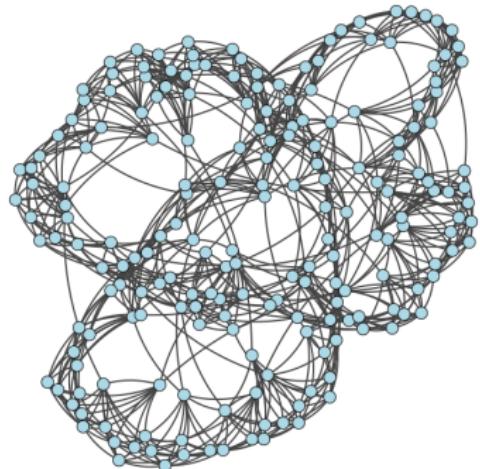
Random walk models

- ▶ we can use **random walks** to model paths/walks/trajectories in a network

random walk model

A random walk model is a stochastic model for walks through a state space Ω

- ▶ random variable $X_t \in \Omega$ assumes state of walker at time t
- ▶ sequence X_t is called a random walk
- ▶ time t and state space Ω can be discrete or continuous



trajectory of a random walk in a small-world network

- ▶ for random walk in network $G = (V, E)$ we consider discrete state space $\Omega = V$, where **transitions occur across edges E**
- ▶ important class of models for **stochastic processes** and basis for **machine learning techniques** → Course: Machine Learning for Complex Networks

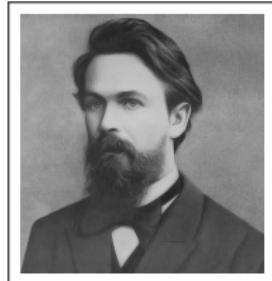
Notes:

- What we have considered is a specific type of a **random walk model**, a popular class of stochastic models which can, among other things, be used to model diffusion processes in networks.
- Referring to our illustrative problem, your drunk friend is modelled by a “random walker” who starts in some initial “state” and who then random walks through the “state space”. In our case, the states of the process are locations in Würzburg, and possible transitions occur between locations that are directly connected by a street.
- We obtain a sequence of states that can be described by a random variable X_t , where X_t is the state of the random walk at time t . In the network context, each random realisation of this sequence X_t is a random path through the nodes of the network, generated by a random walk process.
- As we will see in a moment, this provides us with a simple method to calculate, not only the probability of paths, but also the probability that our random walker **resides** at any given node at any given point in time.

Random walks as Markov chains

- ▶ consider a **discrete-time** random walk on finite and **discrete state space** Ω where $X_t \in \Omega$ is the state at time t
- ▶ at time t **transition probability** of random walk to s_{t+1} is

$$P(X_{t+1} = s_{t+1} | X_t = s_t, X_{t-1} = s_{t-1}, \dots, X_0 = s_0)$$



Andrey Markov

1856 – 1922

- ▶ process is **memoryless** (has the Markov property) iff

$$P(X_{t+1} = s_{t+1} | X_t = s_t) = P(X_{t+1} = s_{t+1} | X_t = s_t, X_{t-1} = s_{t-1}, \dots, X_0 = s_0)$$

for all sequences of states $s_0, \dots, s_{t-1} \in \Omega$

observations

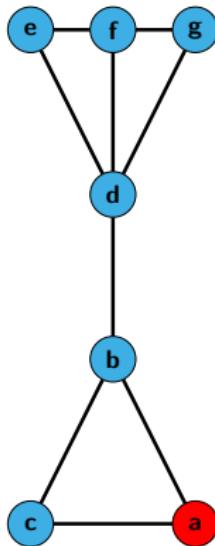
- ▶ future state X_{t+1} only depends on current state X_t
- ▶ discrete-time/discrete-state stochastic process with Markov property is called **Markov chain**
- ▶ transition probabilities between i and j can be given as matrix T_{ij}

image credit: Wikimedia Commons, public domain

Notes:

- But how exactly do we model how the random walker moves between nodes? Since we don't know the exact path generated we assume a simple probabilistic model where each next state is randomly chosen from the set of possible next states. That is, at each time t the choice where to move next is made by means of a random experiment, where we assign transition probabilities to all subsequent states s_{t+1} .
- We can think of numerous ways to assign those probabilities, each capturing a different "model" of the underlying process. In general, the probability of a transition to the next state can depend on the whole history of the process, i.e. where the process has been in each of the prior steps.
- A particularly simple (and well-studied) class of random walk models are those which are "memoryless". This means we assume that the probabilities for transitions to the next state only depend on the current state. In other words we assume that – as soon as a random walker arrives at state s_t – it has "forgotten" where it was before. Such a random walk is a primary example for a so-called **memoryless Markov process**, i.e. a class of processes where the future trajectory of the process is independent of the past. In this case, we can simply describe the transition probability of the process by a simple condition probability.
- If both time and state space are discrete we obtain a **Markov chain**, named after the Russian mathematician Andrey Markov who studied such processes in the late 19th century.

Adjacency matrices of networks



random walker with $X_0 = a$

$$\mathbf{A} = \begin{bmatrix} a & b & c & d & e & f & g \\ a & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ b & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ c & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ d & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ e & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ f & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ g & 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

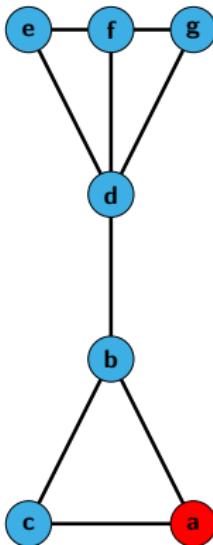
for matrix entry A_{ij} , i refers to row
and j refers to column

how can we define entries of **transition matrix \mathbf{T}** ?

Notes:

- Since the probability to transition to a state j at time $t + 1$ only depends on which state i the process is in at time t , we can define a transition matrix \mathbf{T} with entries T_{ij} being the transition probabilities.
- How can we define the transition matrix of such a random walk process in a network topology, where the states are the nodes in which the random walker currently **resides**? Consider how we have defined the adjacency matrix of a network.
- In our definition of the adjacency matrix, we used the (usual) notation where for entry A_{ij} i refers to the row and j refers to the column.
- Let us now consider a random walker in node (i.e. state) a as illustrated above. The two one entries A_{ab} and A_{ac} correspond to the two possible next states s_{t+1} of a random walker that is in state a at time t .
- How can we define the transition probabilities T_{ij} of such a process?

Transition matrix of a random walk



random walker with $X_0 = a$

$$\mathbf{T} = \begin{bmatrix} a & b & c & d & e & f & g \\ a & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ b & \frac{1}{3} & 0 & \frac{1}{3} & \frac{1}{3} & 0 & 0 \\ c & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 \\ d & 0 & \frac{1}{4} & 0 & 0 & \frac{1}{4} & \frac{1}{4} \\ e & 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ f & 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} & 0 \\ g & 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} \end{bmatrix}$$

T_{ij} captures **probability of walker** residing in node i to move to node j

we can **define entries of a transition matrix** as

$$T_{ij} := A_{ij} \cdot \left(\sum_{k \in V} A_{ik} \right)^{-1} = \frac{A_{ij}}{d_{out}(i)}$$

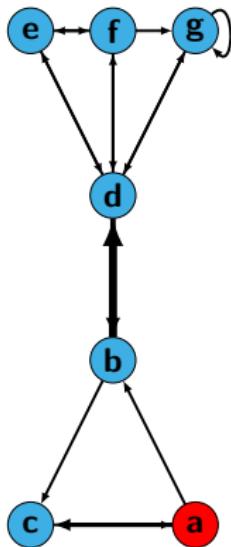
Notes:

- In absence of any further knowledge about the random walker it is reasonable to make a maximum entropy assumption, i.e. we assign the same probability for the transition to each of our neighbours.
- For each entry ij in the adjacency matrix of a directed network, we can simply take the inverse of the out-degree of node i , which counts the number of possible next states (i.e. nodes) that we can transition to. We obtain a transition matrix for a random walk process where in each step the next node is chosen uniformly at random from the set of all possible next nodes.
- In other words: we define a transition matrix \mathbf{T} by dividing the entries of the adjacency matrix \mathbf{A} of a network by the row sums, i.e.

$$T_{ij} = \frac{A_{ij}}{\sum_k A_{ik}}$$

- Note that in this notation, i designates the row and j designates the column of the matrix, i.e. in the example above we have $T_{ab} = \frac{1}{2}$ while $T_{ba} = \frac{1}{3}$.
- This notation implies that all rows of our transition matrix sum up to one (i.e. we have a so-called row-stochastic or right-stochastic matrix, but more on this later).

Biased random walks in weighted networks



$$\mathbf{T} = \begin{bmatrix} a & b & c & d & e & f & g \\ a & 0 & \frac{1}{4} & \frac{3}{4} & 0 & 0 & 0 & 0 \\ b & 0 & 0 & \frac{1}{8} & \frac{7}{8} & 0 & 0 & 0 \\ c & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ d & 0 & \frac{4}{7} & 0 & 0 & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} \\ e & 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ f & 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} & 0 & \frac{1}{3} \\ g & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & \frac{1}{2} \end{bmatrix}$$

in weighted networks we use weights to **bias transition probabilities**, e.g.

$$T_{ij} := A_{ij} \cdot \left(\sum_{k \in V} A_{ik} \right)^{-1} = \frac{w(i, j)}{\sum_{k \in V} w(i, k)}$$

Notes:

- What if we have more information about the random walk process?
- For instance, in a weighted network, we can assume that the random walker is (proportionally) more likely to follow “stronger” links. In our example, we can for instance improve our models by assuming that our drunk friend is more likely to follow broader rather than narrow streets.
- For weighted networks, we can consider non-uniform, **biased transition probabilities** to move to different neighbours. Depending on what we want to model, we can think of many different ways to do this. Here we have used the simplest possible bias in transition probabilities: a bias that is proportional to weights.
 - in the matrix above, the entries $T_{ab} = \frac{1}{4}$ and $T_{ac} = \frac{3}{4}$ are proportional to link weights $w(a, b) = 1$ and $w(a, c) = 3$
 - This implies that we do not **preserve** the absolute weights of links, we normalise the weights for each node individually to obtain proper transition probabilities. Specifically, we can scale the weights of the outgoing links of each node by a constant and still obtain the same transition matrix (because we normalise with the sum of out-weights for each node individually). This implies that many different weighted networks give rise to exactly the same transition matrix.

Stochastic matrices

- row sums in transition matrix \mathbf{T} are one, i.e.

$$\sum_{j \in V} T_{ij} = 1 (\forall i \in V)$$

- such a matrix is called **row-stochastic or right-stochastic**
- let $\pi = (\pi_i)_{i \in V}$ be a **stochastic row vector**, i.e. $\sum_i \pi_i = 1$
- $\pi \cdot \mathbf{T}$ exists and is again a **stochastic row vector**

$$\mathbf{T} = \begin{bmatrix} a & b & c & d & e & f & g \\ a & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ b & \frac{1}{3} & 0 & \frac{1}{3} & \frac{1}{3} & 0 & 0 \\ c & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 \\ d & 0 & \frac{1}{4} & 0 & 0 & \frac{1}{4} & \frac{1}{4} \\ e & 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ f & 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} & 0 \\ g & 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} \end{bmatrix}$$

example

$$\pi = (1, 0, 0, 0, 0, 0, 0)$$
$$\pi \cdot \mathbf{T} = \left(0, \frac{1}{2}, \frac{1}{2}, 0, 0, 0, 0\right)$$

Notes:

- In both cases (weighted and unweighted networks) we obtain a matrix where each row is a probability mass function. Such a matrix is called a **row- or right-stochastic** matrix.
- Why right-stochastic? Because if we multiply \mathbf{T} from the **right** with a stochastic row vector v , i.e. $v \cdot \mathbf{T}$, then the result is again a stochastic row vector, i.e. a vector whose entries sum to one.
- Alternatively, we can also consider **column- or left stochastic** matrices, but then we have to consider column vectors and we need to multiply \mathbf{T} from the left, i.e. we consider $v = (1, 0, 0, 0, 0, 0, 0)^T$ and $\mathbf{P} \cdot v$ returns a stochastic column vector.
- For a network with n nodes, let us take a closer look at a stochastic vector $\pi \in \mathbb{R}^n$ and transition matrix $\mathbf{T} \in \mathbb{R}^{n \times n}$. We can interpret π as a vector that describes the probability of a random walk to **reside** in particular nodes.
- Let us now consider a canonical unit vector π that captures the state of a random walker **residing** in a given node with probability one (see above). What do we get for $\pi \cdot \mathbf{T}$?
- We obtain a vector that captures the probabilities of a random walker to be in a particular node after a single step of the random walk (assuming that it started in the node captured by π).

Visitation probabilities

- ▶ let $\pi^{(t)} = (\pi_1^{(t)}, \pi_2^{(t)}, \dots)$ be **visitation probabilities** at time t
- ▶ we call $\pi^{(0)}$ the **initial distribution** of a random walk

example

$$\pi^{(0)} = (1, 0, 0, 0, 0, 0, 0)$$

$$\pi^{(1)} = \left(0, \frac{1}{2}, \frac{1}{2}, 0, 0, 0, 0\right) = \pi^{(0)} \cdot \mathbf{T}$$

$$\begin{aligned}\pi^{(2)} &= \left(\frac{5}{12}, \frac{1}{4}, \frac{1}{6}, \frac{1}{6}, 0, 0, 0\right) \\ &= \pi^{(1)} \cdot \mathbf{T} = \pi^{(0)} \cdot \mathbf{T}^2\end{aligned}$$

$$\mathbf{T} = \begin{bmatrix} a & b & c & d & e & f & g \\ \begin{matrix} a \\ b \\ c \\ d \\ e \\ f \\ g \end{matrix} & \begin{bmatrix} 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 \\ \frac{1}{3} & 0 & \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{4} & 0 & 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} & 0 & \frac{1}{3} \\ 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \end{bmatrix} \end{bmatrix}$$

using initial distribution $\pi^{(0)}$ and transition matrix \mathbf{T} **visitation probabilities** at time t are given as $\pi^{(t)} = \pi^{(0)} \cdot \mathbf{T}^t$

Notes:

- We can encode the initial state of a random walk process in a stochastic vector $\pi^{(0)}$, which we refer to as the **initial distribution**. We can multiply this vector with the transition matrix and call the result $\pi^{(1)}$.
- Comparing $\pi^{(0)}$ with $\pi^{(1)}$, we observe that the initial probability 1 to be at node a gets “distributed” to probabilities $\frac{1}{2}$ at nodes b and c to whom node a is connected. This is easy to understand, because (i) the transition matrix \mathbf{T} captures the topology of links in the network, and (ii) \mathbf{T} is a right-stochastic matrix, so we again obtain a probability distribution.
- The same applies if we repeatedly multiply the transition matrix with a stochastic vector. The result of k repeated multiplications corresponds to a single multiplication with the k -th power of the transition matrix. That is, for a given transition matrix and a given initial state we can directly calculate the probabilities that the random walker is at any given node after k steps. This is due to the definition of matrix multiplication, where the entries of the k -th power of an adjacency (or transition) matrix capture all paths of exactly length k .
- In the example above, for entries $\pi_b^{(1)}$ and $\pi_c^{(1)}$ we have a difference of $\frac{1}{2}$ compared to $\pi_b^{(0)}$ and $\pi_c^{(0)}$, while for $\pi_a^{(1)}$ we have a difference of 1. All other entries remain unchanged.
- How can we quantify these changes and how do the changes in probability evolve over time?

Total variation distance

- ▶ we can view **evolution of visitation probabilities** as diffusion process

total variation distance

to measure difference between π and π' we define the **total variation distance** as

$$\delta(\pi, \pi') := \frac{1}{2} \sum_i |\pi_i - \pi'_i|$$

- ▶ in our example network, we observe

$$\delta(\pi^{(t)}, \pi^{(t-1)}) \rightarrow 0$$

for $t \rightarrow \infty$

- ▶ in this example random walk process reaches a **stationary state**

example: $t = 15$ vs. $t = 16$

$$\pi^{(15)} \approx (0.124, 0.178, 0.124, 0.212, 0.103, 0.153)$$

$$\pi^{(16)} \approx (0.122, 0.177, 0.122, 0.214, 0.104, 0.156)$$

total variation distance in example

$$\delta(\pi^{(2)}, \pi^{(1)}) \approx 0.583$$

$$\delta(\pi^{(16)}, \pi^{(15)}) \approx 0.007$$

Notes:

- We can calculate a distance measure between two probability mass functions. One possible measure is the so-called **total variation distance**, which simply sums the component-wise (absolute) differences.
- Note that the total variation distance (TVD) is at most 1, because the sum of the differences for each dimension is at most two, and we normalise this maximum value by the multiplication with $\frac{1}{2}$ (see difference between $\pi^{(0)}$ and $\pi^{(1)}$ on previous slide).
- We can now study how the total variation distance between two consecutive visitation probability vectors $\pi^{(t)}$ and $\pi^{(t+1)}$ evolves for increasing t .
- For our simple example we find that the difference decreases over time (in this case it actually converges to zero).
- This means that – if we allow the random walk process to run long enough – the visitation probabilities will not change any more. For our drunk friend walking in Würzburg, this means that (under certain conditions that we will outline in a moment) after a long enough time we can calculate (stable) probabilities to find him or her at any of the possible locations in Würzburg. We could then search at those locations that have the highest visitation probability.
- Similar ideas have actually been used in real search and rescue operations, e.g. in the search for missing ships.

Stationary distribution of random walks

- ▶ in our example $\pi^{(t)}$ ceases to change, i.e. process reaches a **stationary distribution** $\pi := \pi^{(\infty)}$ such that

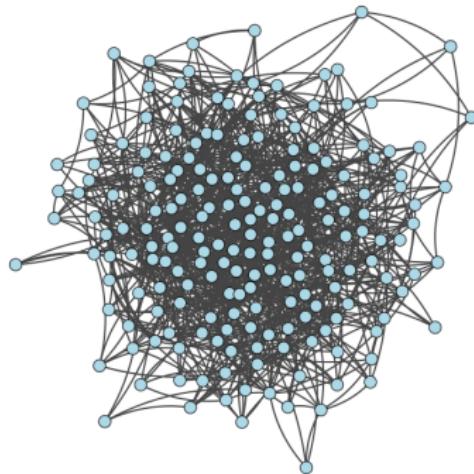
$$\pi = \pi \cdot \mathbf{T}$$

- ▶ this is an **eigenvalue problem** of the form

$$\pi \cdot v = \pi \cdot \mathbf{T}$$

with $v = 1$

- ▶ we can calculate stationary distribution as **left eigenvector of transition matrix corresponding to eigenvalue $v = 1$**



convergence to stationary distribution in an example network

Notes:

- In cases where the visitation probability $\pi^{(t)}$ ceases to change after long enough time, we call the limiting distribution $\pi := \pi^{(\infty)}$ the **stationary distribution** of the random walk process. We can easily calculate this stationary distribution by solving an eigenvector problem for the eigenvalue one of the transition matrix.
- Note that – since the transition matrix T is row-stochastic, i.e. rows sum up to 1 – the eigenvalue $v = 1$ is at the same time the largest eigenvalue of matrix T .
- Also, here we must use the **left eigenvector**, because T is a row- or right-stochastic matrix. If we were to define T as a column- or left-stochastic matrix, we would need to consider right eigenvalues, i.e. we would need to solve the eigenvalue problem:

$$T \cdot \pi = \pi \cdot v$$

- In general, the eigenvalue equation above can have multiple solutions for $v = 1$, i.e. we can have several vectors π that all represent stationary distributions of the random walk process (for different initial distributions $\pi^{(0)}$). We will see that the question whether the eigenvalue 1 occurs only once in the sequence of all eigenvalues of matrix T has an interesting graph-theoretic analogy. Similarly, there can be the case that there is no solution to the eigenvector equation, in which case visitation probabilities do not converge.

Practice Session

- ▶ we show how we can simulate random walks in networks using python
- ▶ we calculate the total variation distance between stochastic vectors
- ▶ we use eigenvectors to calculate stationary visitation probabilities in a given network

observation

stationary distribution does not exist for all networks!

practice session

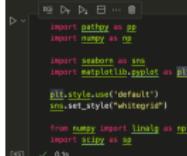
see notebook 11-01 in gitlab repository at

→ https://gitlab.informatik.uni-wuerzburg.de/ml4nets_notebooks/2022_wise_sna_notebooks

10-01: Simulating Random Walks in Networks

January 12 2023

In the first unit we explore random walks, a simple yet powerful stochastic model for diffusion processes in networks. Random walk processes are not only important as a model for diffusion. Their stationary distribution can also be used to define centrality measures based on the eigenvectors of the transition matrix. As we shall see in the subsequent course "Machine Learning for Complex Networks", random walks (and more generally Markov chains) are also a powerful and widely applied framework for statistical learning in networks, with applications in node ranking, link prediction, representation learning, and graph clustering.

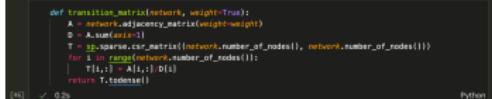


```
import networkx as nx  
import numpy as np  
  
import seaborn as sns  
import matplotlib.pyplot as plt  
  
plt.style.use('default')  
sns.set_style("whitegrid")  
  
from numpy import linalg as np_linalg  
import scipy as sp
```

Python

Computing Transition matrices

We first write a function that computes a (row- or right-stochastic) transition matrix of a random walk process for a given network. The method should work for weighted, unweighted, directed, and undirected networks represented by a `networkx` object.



```
def transition_matrix(network, weight=True):  
    """Compute the transition matrix (weight=weight)  
    G = A.sum(axis=1)  
    T = sp.spmatrix((network.number_of_nodes(), network.number_of_nodes()))  
    for i in range(network.number_of_nodes()):  
        T[i,:] = A[i,:]/G[i]  
    return T.todense()
```

Python

Notes:

- In the second practice session, we experimentally test the conditions under which the Markov chain convergence theorem holds, i.e. we test for the existence and uniqueness of the stationary distribution in networks with different topologies.
- Interestingly, our results show that we can use the sequence of eigenvalues of a transition matrix to make statements about the topology of a network, namely how many (strongly) connected components it contains.
- We will explore this in more detail in the next lecture.

Stationary distribution of random walks

- ▶ stationary distribution of random walks is an **important quantity** in network analysis and data mining
- ▶ we can use it to define a **node centrality measure** → Exercise 08

random-walk based centrality

- ▶ let random walker “explore” a network
- ▶ we expect “central” nodes to be visited more frequently
- ▶ for stationary distribution π define centrality of node v as π_v
- ▶ when does a **unique** stationary distribution **exist?**



image credit: screenshot of Google search, self-created

Notes:

- Why do we even consider the stationary distribution of a random walk process?
- Apart from being potentially useful in search operations, it actually rise to an interesting notion of node centralities in networks. The idea is that we define a random walk process, initialise it at some node, and then let it explore the network topology.
- Intuitively, the probability to find the random walker – after a long enough time – in a given node v should correspond to the “centrality” of v in the topology. This was actually a key idea behind the first prototype of the Google search engine back in the mid 1990s (more on this later).
- But before being able to interpret the stationary distribution in this way we must first answer two more fundamental questions: when does a stationary distribution exist and when is it unique?
- Think about this problem for a moment:
 - Can you think of a network where no stationary distribution exists?
 - Can you think of a network where two different stationary distributions exist?

Existence of a stationary distribution

- ▶ for a given network, **does a stationary distribution exist** for every possible initial distribution π_0 ?

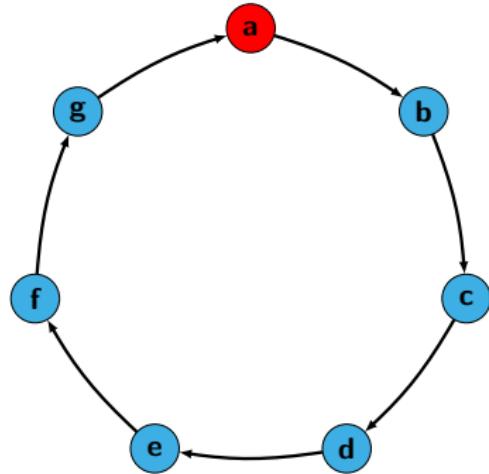
example: directed ring lattice

$$\pi^{(0)} = (1, 0, 0, 0, 0, 0, 0)$$

$$\pi^{(1)} = (0, 1, 0, 0, 0, 0, 0)$$

$$\pi^{(999)} = (0, 0, 0, 0, 0, 1, 0)$$

$$\pi^{(1000)} = (0, 0, 0, 0, 0, 0, 1)$$



- ▶ in this example network, we observe **periodic behavior**

example

network in which a random walk does not reach a stationary distribution for
 $\pi^{(0)} = (1, 0, \dots, 0)$

Notes:

- We first study the **existence of a stationary distribution** in a simple toy example of a directed ring network where each node is only connected to exactly one neighbour via a directed link.
- We further assume an initial distribution $\pi^{(0)}$ capturing that the random walker is initially at node a (with probability 1).
- Trivially, in this case there is no stationary distribution. We observe a periodic behaviour that is due to the fact that the topology forces the random walker to “circle” through the nodes.
- Thus, this network is an example for which a stationary distribution does not exist for **all initial distributions**. Specifically it does not exist for the $\pi^{(0)}$ above. But we can still find an initial distribution for which there is a stationary distribution. This is actually the case if all initial visitation probabilities are the same, i.e. $\pi_i = \frac{1}{7}$. Here this is the only initial distribution for which a stationary state is reached.
- So the network above has no stationary distribution for all initial distributions (but one). We can intuitively relate this property to the directed ring topology of the network. Would anything change if we were to make the ring undirected? How can we capture this property of a network formally (and how can we avoid it)?

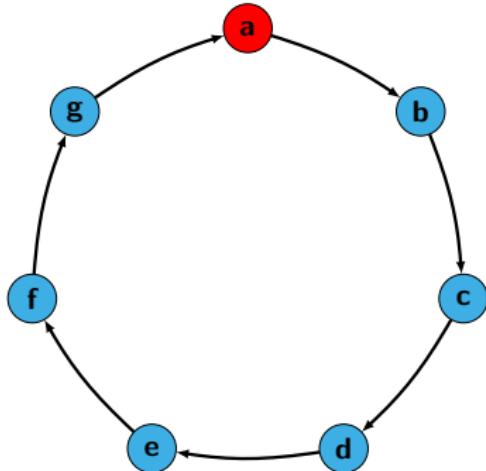
Aperiodicity

- ▶ let $\gcd \{a_1, \dots, a_n\}$ be the **greatest common divisor** of numbers a_1, \dots, a_n
- ▶ for state i of a Markov chain, we define the **period $r(i)$** of state i as

$$r(i) = \gcd \{t \geq 1 : (\mathbf{T}^t)_{ii} > 0\}$$

i.e. gcd of times t when walk can return to i

- ▶ we call transition matrix **aperiodic** iff $r(i) = 1 \forall i \in V$
- ▶ transition matrix is **periodic** if $\exists i \in V : r(i) \neq 1$
- ▶ can we **make this network aperiodic?**



example

$r(i) = 7$ for all $i \in V$, i.e. transition matrix \mathbf{T} of a random walk in this network is periodic

Notes:

- The circular structure of the topology translates to the property of **periodicity** of the Markov chain that describes the random walk process.
- We can ask a simple question: At which times can a random walker starting at node i possibly return to node i , i.e. for which times t is $(\mathbf{T}^t)_{ii} > 0$? We can then define $r(i)$ as the greatest common divisor of all times t at which the the random walker can return to node i .
- In the example above, it is easy to see that entries T_{ii}^t of the t -th power of the transition matrix \mathbf{T} are 1 (and thus larger than zero) for $t = 7, t = 14, t = 21, \dots$. This implies that $r(i) = 7$ for all $i \in V$.
- We call a transition matrix in which the greatest common divisor of all times t at which $(\mathbf{T}^t)_{ii} > 0$ is one **aperiodic**. Otherwise, we call it **periodic**.
- We will use the statements “the transition matrix is periodic/aperiodic” and “the network is periodic/aperiodic” interchangeably, since we can always map a network to a transition matrix and vice-versa.
- Q: do you have an idea how we could make this network aperiodic? In other words: how can we trivially make the greatest common divisor 1 for all states?
- A: if we simply add a positive probability to stay in a state, we trivially have $(\mathbf{T}^t)_{ii} > 0$ for $t = 1$. Remember that $\gcd(1, n) = 1$ for any $n \geq 1$

Lazy random walks

- ▶ we can make **any** transition matrix aperiodic by setting

$$T_{ii} = \epsilon$$

for all $i \in V$ and $\epsilon > 0$

- ▶ we obtain a **lazy random walk**, where the walker may rest in each step
- ▶ corresponds to random walk in **network with self-loops**

$$\mathbf{T} = \begin{bmatrix} a & b & c & d & e & f & g \\ \frac{1}{100} & \frac{99}{100} & 0 & 0 & 0 & 0 & 0 \\ b & 0 & \frac{1}{100} & \frac{99}{100} & 0 & 0 & 0 \\ c & 0 & 0 & \frac{1}{100} & \frac{99}{100} & 0 & 0 \\ d & 0 & 0 & 0 & \frac{1}{100} & \frac{99}{100} & 0 \\ e & 0 & 0 & 0 & 0 & \frac{1}{100} & \frac{99}{100} \\ f & 0 & 0 & 0 & 0 & 0 & \frac{1}{100} \\ g & \frac{99}{100} & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

example

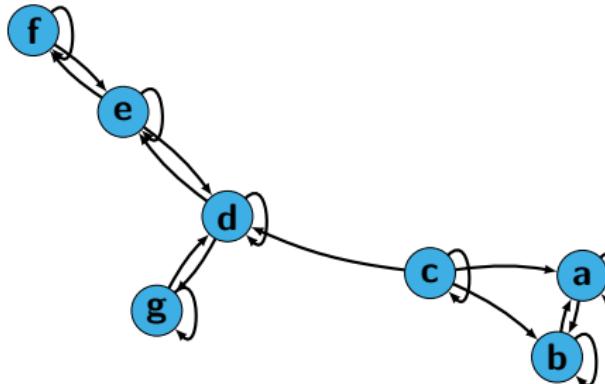
by setting $T_{ii} > 0$ for all $i \in V$ we can make \mathbf{T} of random walk in ring lattice network aperiodic

Notes:

- This leads us to the interesting result that we can make any transition matrix aperiodic, simply by adding a non-zero probability that the random walk **stays** in the node where it currently resides. A random walk model where the walker can simply stay in the current node is called a lazy random walk.
- From a network perspective, this property translates to the addition of (possibly weighted) self-loops.
- Lazy random walks are of great theoretical importance since we often want to make analytical statements about diffusion speed or eigenvectors of certain transition matrices (extracted, e.g., from an empirical network).
- In these cases, adding a small self-loop probability ϵ excludes periodic transition matrices and thus guarantees the convergence of a random walk process, i.e. we ensure that the right eigenvalue of \mathbf{T} exists.

Uniqueness of stationary distribution

if a stationary distribution exists, is it **unique** for all initial distributions $\pi^{(0)}$?



example 1

random walk starts in node a , i.e.

$$\pi^{(0)} = (1, 0, 0, 0, 0, 0, 0)$$

$$\pi^{(50)} \approx \left(\frac{1}{2}, \frac{1}{2}, 0, 0, 0, 0, 0\right)$$

example 2

random walk starts in node g , i.e.

$$\pi^{(0)} = (0, 0, 0, 0, 0, 0, 1)$$

$$\pi^{(50)} \approx \left(0, 0, 0, \frac{3}{10}, \frac{3}{10}, \frac{2}{10}, \frac{2}{10}\right)$$

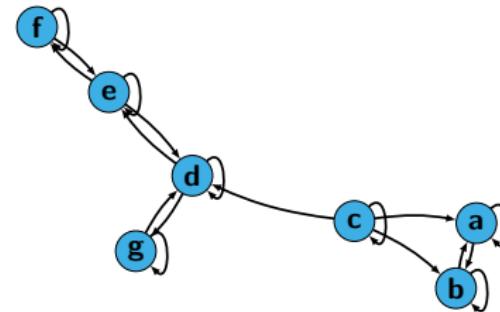
for this network, random walk has **no unique** stationary distribution

Notes:

- Let us now consider the second question: Given that a stationary distribution exists we ask whether there is only one unique stationary distribution independent of the initial distribution that we started with.
- The simple toy example above shows that this is – in general – not the case. Here it actually matters where we started: If we were to start in a or b , we can only visit the left part of the network (i.e. a and b). If we start in $d - g$, we will always stay in the right part of the network (i.e. $d - g$)
- What happens if we start in node c ? In this case, the visitation probabilities converge to a third stationary solution $\left(\frac{1}{3}, \frac{1}{3}, 0, \frac{1}{10}, \frac{1}{10}, \frac{1}{15}, \frac{1}{15}\right)$
- We observe that the existence of multiple stationary distributions is due to the fact that certain parts of this (directed) network are not connected via paths (e.g. there is no path from node d to a).

Irreducible transition matrices

- ▶ transition matrix \mathbf{T} is **reducible** if a δ exists such that $(\mathbf{T}^t)_{ij} = 0$ for a pair i, j and all $t > \delta$, otherwise it is called **irreducible**
- ▶ irreducibility maps to **connectivity** of networks
- ▶ transition matrix of any **directed and strongly-connected network** is irreducible
- ▶ transition matrix of any **undirected connected network** is irreducible



example for reducible transition
matrix for weakly connected network

Notes:

- This property of a network is captured by the **reducibility** of the transition matrix. We call a transition matrix reducible, if there exists at least one pair of nodes i, j where no walk from i is to j is possible for any t larger than some threshold ϵ .
- the term “reducible” refers to the fact that we can still study random walk processes **reduced** to subsets of nodes and links where walks exist between any pair in the subset. These subsets correspond to “strongly connected components”, i.e. the maximal subsets of nodes such that all nodes in the subset are connected by paths in both directions.
- In the example above, we have three strongly connected components consisting of $a - b, d - g$ and c , each of them corresponding to one of the three different stationary distributions of the random walk.
- The existence of multiple different stationary distributions (i.e. multiple eigenvectors of \mathbf{T}) in this example implies that also the eigenvalue 1 appears multiple times (because now the eigenvalue problem has multiple solutions). This implies that the eigenvalue sequence of a transition matrix tells us something about the topology of the underlying network (more on this next week).
- In undirected networks, the notions of connectivity and strong connectivity coincide. Directed networks can be weakly connected but not strongly connected iff the following two conditions hold simultaneously: (i) for all pairs of nodes i, j there exists a path from i to j or from j to i and (ii) for at least one pair i, j either a path from i to j or from j to i does not exist.

Markov chain convergence theorem

Markov chain convergence theorem

Let \mathbf{T} be an **irreducible and aperiodic transition matrix** and let $\pi^{(0)}$ be an **arbitrary initial distribution**.

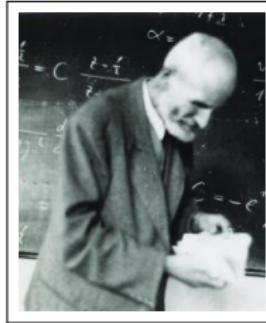
For $\pi^{(t)} := \pi^{(0)} \cdot \mathbf{T}^t$ there exists **exactly one** stationary distribution π such that

$$\delta(\pi^{(t)}, \pi) \rightarrow 0 \quad (t \rightarrow \infty)$$

- ▶ follows from **Perron-Frobenius theorem** for real-square matrices with positive entries

observation

- ▶ if conditions of theorem are satisfied the stationary distribution π is **independent from initial distribution**
- ▶ random walker “forgets” where it started (information decreases/entropy increases)



Oskar Perron
1880 – 1975



Georg Frobenius
1849 – 1917

image credit: Konrad Jacobs,
Wikimedia Commons, CC BY-SA
2.0

image credit: Oberwolfach
Photo Collection, public
domain

Notes:

- Combining the properties of aperiodicity and irreducibility of a Markov chain, we can obtain a central result in Markov theory: the Markov chain convergence theorem, which clarifies the conditions under which a Markov chain has a unique stationary distribution.
- The Markov chain convergence theorem follows as a special case from the Perron-Frobenius theorem for non-negative (i.e. all values are either zero or positive) and irreducible matrices where all states have period 1, see details in → O Perron, 1907 and → G Frobenius, 1912
- We omit the proof of the theorem, but a nice in-depth coverage of Markov chains with easy to follow proofs (e.g. of the Markov chain convergence theorem), illustrative examples and applications can be found in → O Häggström, 2002
- An important implication of the theorem is that, if the assumptions of the theorem are satisfied, the stationary distribution—since it is unique—is independent of the initial distribution. This implies that the random walker, over time, forgets in which node it started. This is a result of the memoryless/Markov property of the process, by which we “destroy information” on our origin in each step of the process. Note that a single step of the process does not destroy all information about the origin, since from the current location we can still say from which nodes we could possibly have started. This information will fade over time.

Practice Session

- ▶ we explore the existence of a stationary distribution
- ▶ we explore the uniqueness of the solution to the eigenvalue equation in networks with multiple strongly connected components

observation

we can use eigenvalues of matrix representations of a network to make statements about its topology

10-02: Stationary States of Random Walks

January 12 2022

In the second unit we study the stationary state of random walks in networks. With this we will lay the foundation for the analytical study of diffusion speed in the upcoming lecture.

```
import astropy as ap
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

plt.style.use('default')
sns.set_style('whitegrid')

from numpy import linalg as nl
import scipy as sp
```

[2] ✓ 0:0s + Code + Markdown Python

```
def transition_matrix(network, weight=True):
    A = network.adjacency_matrix(weight=weight)
    D = A.sumaxis(0)
    T = np.diag(D) * matrix(network.number_of_nodes(), network.number_of_nodes())
    for i in range(network.number_of_nodes()):
        T[i,i] = A[i,i]/D[i]
    return T.toarray()

def visitation_probability(network, initial_dist, t):
    T = transition_matrix(network)
    p_t = np.dot(initial_dist, np.linalg.matrix_power(T,t))
    return p_t

def stationary_distribution(network):
    T = transition_matrix(network)
    eigenvalues, eigenvectors = np.linalg.eig(T, left=True, right=False)
    # With the following line we take the +leading eigenvector+, i.e. the eigenvector
    # corresponding to the +largest eigenvalue+ of one
    # np.argmax returns an array of indices that would sort an array
    # in multidimensional slicing [x], which returns all entries
    # in column x
    pi = eigenvectors[:,np.argmax(eigenvalues)].T[0]
    # We made sure that the eigenvector is normalised
    pi = pi/sum(pi)
    return pi
```

[2] ✓ 0:0s Python

practice session

see notebook 10-02 – 10-03 in gitlab repository at

→ https://gitlab.informatik.uni-wuerzburg.de/ml4nets_notebooks/2022_wise_sna_notebooks

Notes:

- In the practice session, we experimentally study the relative size of the surviving connected component of scale-free networks under a random failure model. We find that the surviving connected component size decreases linearly with the failure probability q . This is at odds with our theoretical prediction. We would expect that for a scale-free network with exponent between 2 and 3, the size of the largest surviving connected component is close to one.
- The reason for this difference between theory and reality is the finite size of (real) networks. In other words: for a network with exponent $\gamma = 2.5$ the prediction (which we have obtained in the limit of infinite networks) is not a good approximation for a finite network with 1000 nodes. We have seen that this finite size was not an issue for Erdős-Rényi networks, but it is an issue for scale-free networks.

Exercise sheet 08

- ▶ eighth exercise sheet is available on WueCampus
 - ▶ study random walks in real networks
 - ▶ consider visitation probabilities as node centralities
 - ▶ study diffusion speed in networks
- ▶ solutions are due January 18th (via WueCampus)
- ▶ present your solution to earn bonus points



Exercise Sheet 08

Published: January 11, 2022

Due: January 19, 2022

Total points: 10

1. Basics of Markov chain models

Consider a graph $G = (V, E)$ with the following adjacency matrix:

$$A = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 2 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 2 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(a) Assuming that the row and column numbers correspond to nodes (enumerated from 1 to 5), calculate the probability that a random walker starting in node 1 will traverse the following sequence of nodes: (1, 2, 3, 2, 4, 1, 5, 1, 2).

(b) Does a unique stationary distribution exist for a random walk in graph G ? If yes, calculate the stationary distribution, if not explain why.

(c) What is the relative frequency at which we expect the sequence of nodes (1, 2) to appear in an infinite random walk on graph G ?

2. Random Walks and Node Centralities

Consider a (small) directed network in the `netschule` data. Check whether a unique stationary distribution exists. If necessary ensure that the network is aperiodic.`

(a) Compute the stationary visitation probabilities of nodes and rank the nodes based on those probabilities. Compare the resulting ranking with a ranking based on the in- or out-degree of nodes or other path-based measures. Explain what you observe.

(b) Make the network undirected and compare the ranking of nodes based on stationary visitation probabilities to a degree-based ranking. Can you explain your observation in an analytical way?

3. Diffusion speed in networks

Consider microstates generated by the (one-dimensional) Watts-Strogatz model with parameters $n = 100$, $\alpha = 3$, and three rewiring probabilities $p = 0$, $p = 0.1$, and $p = 1$.

(a) Use the transition matrix to compute the stationary distributions π of a random walk on these three microstates. Hint: Use `numpy` or `scipy` functions to calculate eigenvectors.

(b) Assume that $\pi^{(0)}$ is a random initial distribution of a random walk, where we assign probability one to a randomly chosen node. We further assume that $\pi^{(t)}$ denotes the visitation probabilities of nodes after t steps. Write a python function that computes the total variation distance between two distributions π^t and $\pi^{(t)}$. Use your function to calculate the total variation distances $\|\pi^t - \pi^{(t)}\|$ between the stationary distribution and the visitation probabilities after t steps for different values of t and 50 different random initial distributions. Repeat your experiment for the three microstates mentioned above and plot the evolution of the average total variation distance to the stationary distribution over time.

In which network is the speed of convergence to the stationary distribution largest? Which one exhibits the slowest convergence speed?

Notes:

Self-study questions

1. How is the transition matrix of a random walk defined for weighted and unweighted networks?
2. When does a random walk in a network converge to a unique stationary distribution?
3. How can we compute the stationary distribution of a random walk?
4. How is total variation distance defined? What is its maximum value?
5. For a right-stochastic transition matrix \mathbf{T} and an initial distribution $\pi^{(0)}$, how can we compute visitation probabilities at time t of a random walk?
6. Can you give an example for a network in which a random walk never reaches a stationary distribution?
7. Can you give an example for a network in which a random walk reaches multiple stationary distributions for different initial distributions?
8. How can a periodic transition matrix be made aperiodic?

Notes:

References

reading list

- ▶ A Arenas et al: **Synchronization in complex networks**, Physics Reports, 2008
- ▶ A Barrat, M Barthelemy, A Vespignani: **Dynamical Processes on Complex Networks**, Cambridge University Press, 2008
- ▶ SH Strogatz: **From Kuramoto to Crawford: exploring the onset of synchronization in populations of coupled oscillators**, Physica D: Nonlinear Phenomena 143, 2000
- ▶ AA Markov: **Extension of the limit theorems of probability theory to a sum of variables connected in a chain**, reprinted in Appendix B of R. Howard: *Dynamic Probabilistic Systems, volume 1: Markov Chains*, 1971
- ▶ O Perron: **Zur Theorie der Matrices**, Math. Ann. 64, 1907
- ▶ G Frobenius: **Über Matrizen aus nicht negativen Elementen**, Berl. Ber., 1912
- ▶ O Häggström: **Finite Markov Chains and Algorithmic Applications**, 2002

248

OSCAR PERRON.

Zur Theorie der Matrices.

Von

OSCAR PERRON in München.

In dieser Note werden zum Teil bekannte Sätze aus der Theorie der Matrices und ihrer charakteristischen Gleichung auf neuem, höchst einfache Weise bewiesen, zum Teil neue Sätze entwickelt. Unter den Anwendungen der Theorie habe ich ein der Gräfischen Methode analoges Verfahren zur näherungsweisen Berechnung der Wurzeln einer algebraischen Gleichung hervor.

§ 1.

Sei
$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}$$

das Koeffizientensystem einer linearen Substitution von n Variablen, E das der identischen Substitution. Soweit im folgenden bloß solche Koeffizientensysteme oder kürzer *Matrices* vorkommen, die rationale Funktionen von A , also miteinander vertauschbar sind, dürfen alle rationalen Rechenoperationen genau wie bei Zahlen ausgeführt werden. Bezeichnet man die Elemente des Potenz A^t mit $a_{ij}^{(t)}$, so daß also $a_{ij}^{(t)}$ mit a_{ij} gleichbedeutend ist, so ist wegen $A^{t+s} = A^t A^s$:

$$(1) \quad a_{ij}^{(t+s)} = \sum_{k=1}^n a_{ik}^{(t)} a_{kj}^{(s)}.$$

Konssequenterweise sollen auch die Elemente der Einheitssubstitution $E = A^0$ mit $a_{ij}^{(0)}$ bezeichnet werden; es ist also

$$a_{ij}^{(0)} = \begin{cases} 0 & \text{für } i \neq j, \\ 1 & \text{für } i = j, \end{cases}$$

und (1) gilt auch noch, wenn v oder μ oder beide gleich Null sind.

Sind $A = (a_{ij})$, $B = (b_{ij})$, $C = (c_{ij})$ beliebige Matrices, so sind die Gleichungen

$$A = B \quad \text{oder} \quad A = BC$$

Notes: