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 12
Spectral Analysis of Networks

January 25, 2023

Notes:

- **Lecture L12:** Spectral Analysis of Networks 25.01.2023
- **Educational Objective:** We study how the topology of a network influences diffusion processes. We show how the eigenvalue spectrum of a transition matrix allows us to predict the diffusion speed, and how the spectrum of the Laplacian matrix yields insights about components and cluster structures.
    1. Random Walks and Diffusion Speed in Networks
    2. Continuous-time Diffusion Speed
    3. Laplacian Spectrum of Networks
- **Exercise 10:** Random Walks and Diffusion Speed due 01.02.2023

Motivation

random walks in networks

- we model diffusion of random walk visitation probabilities
$$\pi^{(t+1)} = \pi^{(t)} \cdot \mathbf{T}$$
- stationary state $\pi^{(\infty)} = \pi^{(\infty)} \cdot \mathbf{T}$ can be calculated based on **eigenvector of transition matrix**
- unique stationary state $\lim_{t\to\infty} \pi^{(t)} = \pi$ exists iff transition matrix is **aperiodic and irreducible**
- stationary states of random walks = measure for **feedback centrality**

open questions

1. how long does it take until we reach the stationary state, i.e. when does the diffusion process reach **equilibrium**?
2. how can we analytically predict **diffusion speed** in a given network?
3. what does diffusion speed tell us about the network?

diffusion speed in Watts-Strogatz networks

minimum number of random walk steps $t(\epsilon)$ such that $\delta(\pi^{(t)}, \pi) < \epsilon = 0.005$ for $t > t(\epsilon)$ in Watts/Strogatz network with $n = 200$, $m = 5$ and different rewiring probabilities p (x-axis)

Notes:

- In the last weeks we explored random walks and interpreted the change of visitation probabilities of nodes over time as **diffusion of a conserved quantity** in a network. A different view on this: for a row i, the entries T_{ij}^t give the probability that – assuming a random walk that starts in node i – the walker ends up in j after exactly t steps. In the stationary state, the initial state has been "forgotten", i.e. for large t, $\mathbf{T}^{(t)}$ consists of identical rows, each approaching π.
- We studied under which conditions visitation probabilities converge to a (unique) stationary distribution and how such stationary states can be interpreted as feedback centrality.
- Today we study how fast diffusion in a given network is. Based on the stationary state π, we can calculate how far $\pi^{(t)}$ is away from π. Calculating $\pi^{(t)}$ over time allows us to assess the **speed of diffusion**. This can be viewed as diffusion process that reaches a equilibrium state. We can experimentally explore this using microstates of a network model. For the figure above, we have calculated the minimal number of random walk steps $t(\epsilon)$ needed until visitation probabilities reach a total variation distance from the stationary state that is smaller than $\epsilon = 0.005$. We repeated this for different microstates generated using a Watts-Strogatz model with different rewiring probabilities p (x-axis) and an identical number of nodes and links.
- We find that the topology of the network has a major impact on diffusion speed. For a random graph ($p = 1$) diffusion speed is two orders of magnitude larger than for a ring lattice ($p = 0$). We want to analytically understand and predict how the topology influences diffusion dynamics.

Eigendecomposition of matrices

- let $\mathbf{A} \in \mathbb{R}^{n \times n}$ be a square matrix with non-degenerate **eigenvalues** $|\lambda_1| > |\lambda_2| > \ldots > |\lambda_n|$ and linearly independent **eigenvectors** $\vec{v_1}, \vec{v_2}, \ldots, \vec{v_n}$ (row vectors)
- we define an **eigenmatrix** $\mathbf{U}$ as

$$\mathbf{U} := (\vec{v_i})_{i=1,\ldots,n}$$

- we further define a **diagonal matrix** $\mathbf{D}$ as

$$\mathbf{D} := \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \lambda_n \end{pmatrix}$$

- this yields an **eigendecomposition** or **factorization** of $\mathbf{A}$ as

$$\mathbf{A} = \mathbf{U}^{-1}\mathbf{D}\mathbf{U}$$

Notes:

- Before we can study diffusion speed, we need a quick recap of basic linear algebra. We particularly introduce an essential approach to analytically study diffusion speed: the **eigendecomposition of a matrix**.
- Consider a square matrix $\mathbf{A}$ and let us assume that it has n non-degenerate (i.e. different) eigenvalues and linearly independent eigenvectors. In this case, we can find a so-called (spectral) eigendecomposition (or matrix factorization) of $\mathbf{A}$, i.e. we can express $\mathbf{A}$ as a product of its eigenvectors and eigenvalues as shown above. A variant of this (non-negative matrix factorization) is actually a popular machine learning method, e.g. to perform unsupervised classification.
- Unfortunately, the eigenvectors of a square matrix are not necessarily linearly independent and in these cases the inverse $\mathbf{U}^{-1}$ of the eigenmatrix may not exist. However, in such cases we can still compute an approximate eigendecomposition of $\mathbf{A}$ by using a so-called Moore-Penrose pseudo-inverse. For such pseudo-inverse (or generalised inverse) matrices, we have $\mathbf{U}\mathbf{U}^{-1} \approx \mathbf{I}$. Peudo-inverses are an important tool to find approximate solutions to systems of linear equations that do not have an exact solution. $\longrightarrow$ R Penrose, 1956 $\longrightarrow$ EH Moore, 1920
- In the following discussion we thus drop the requirement that eigenvectors are linearly independent, assuming that in these cases we use the Moore-Penrose pseudo-inverse of the eigenmatrix to obtain an approximate eigendecomposition.

Eigendecomposition of transition matrix

- consider transition matrix $\mathbf{T} \in \mathbb{R}^{n \times n}$ of a random walk with stationary distribution π
- let $\vec{v_1}, \ldots, \vec{v_n}$ be eigenvectors and $1 = |\lambda_1| > |\lambda_2| > \ldots > |\lambda_n|$ eigenvalues of $\mathbf{T}$
- **eigendecomposition of $\mathbf{T}$** is given as

$$\mathbf{T} = \mathbf{U}^{-1}\mathbf{D}\mathbf{U}$$

  with eigenmatrix $\mathbf{U}$ and diagonal matrix $\mathbf{D}$ as defined before
- for any **initial distribution** $\pi^{(0)}$ we can write **visitation probabilities** $\pi^{(t)}$ as

$$\pi^{(t)} = \pi^{(0)}.\mathbf{T}^t = \pi^{(0)}.\left(\mathbf{U}^{-1}\mathbf{D}\mathbf{U}\right)^t = \pi^{(0)}\mathbf{U}^{-1}\mathbf{D}\underbrace{\mathbf{U}\mathbf{U}^{-1}}_{\mathbf{I}}\mathbf{D}\mathbf{U}\ldots = \pi^{(0)}.\mathbf{U}^{-1}\mathbf{D}^t\mathbf{U}$$

- with $\vec{a} := \pi^{(0)} \cdot \mathbf{U}^{-1}$ we can express visitation probabilities as

$$\pi^{(t)} = \vec{a} \cdot \mathbf{D}^t\mathbf{U}$$

Notes:

- How does the spectral eigendecomposition help us to study the speed of diffusion processes?
- We can apply this transformation to the transition matrix in order to get a surprisingly simple and strong result about the speed of diffusion processes in arbitrary network topologies.
- Following the analysis above we see that, using the eigendecomposition of the transition matrix, the visitation probabilities after t steps can be expressed based on (i) the eigenmatrix, (ii) the eigenvalues (stored in the diagonal elements of $\mathbf{D}$), and the projection $\vec{a}$ of the initial distribution $\pi^{(0)}$ in the eigenspace of $\mathbf{T}$ (which we obtain by multiplying $\pi^{(0)}$ with the inverse of the eigenmatrix)
- Q: why is $\mathbf{T}^t = \mathbf{U}^{-1}\mathbf{D}^t\mathbf{U}$
- A: we can verify this for $t = 2$ (general case trivially follows by induction):

$$\mathbf{T}^2 = \left(\mathbf{U}^{-1}\mathbf{D}\mathbf{U}\right)\left(\mathbf{U}^{-1}\mathbf{D}\mathbf{U}\right)$$

- We can change the bracketing to

$$\mathbf{U}^{-1}\mathbf{D}\left(\mathbf{U}\mathbf{U}^{-1}\right)\mathbf{D}\mathbf{U} = \mathbf{U}^{-1}\mathbf{D}\mathbf{D}\mathbf{U} = \mathbf{U}^{-1}\mathbf{D}^2\mathbf{U}$$

Evolution of probabilities in eigenspace

- for $\mathbf{D} = \text{diag}(\lambda_1, \ldots, \lambda_n)$ and eigenmatrix $\mathbf{U}$ of $\mathbf{T}$ we found

$$\pi^{(t)} = \vec{a} \cdot \mathbf{D}^t \mathbf{U}$$

- for eigenvectors $\vec{v_i}$ and $\vec{a} = (a_i)_{i=1,\ldots,n}$ we can rewrite $\pi^{(t)}$ as

$$\pi^{(t)} = \sum_{i=1}^{n} a_i \lambda_i^t \vec{v_i}$$

- **aperiodic and irreducible stochastic matrix**: $\lambda_1 = 1$ and $a_1 \vec{v_1} = \pi$, i.e.

$$\pi^{(t)} = \pi + \sum_{i=2}^{n} a_i \lambda_i^t \vec{v_i}$$

- time-evolving **difference to the stationary distribution** is given by

$$\pi^{(t)} - \pi = \sum_{i=2}^{n} a_i \lambda_i^t \vec{v_i}$$

Notes:

- The vector $\vec{a}$ is defined based on the projection of the initial distribution into the eigenspace of the transition matrix. How does this help us to predict diffusion speed?
- We can use the relation $\pi^{(t)} = \pi^{(0)} \cdot \mathbf{T}^t$ and we can again express this in the eigenspace of the transition matrix
- Since $\vec{v_i}$ are eigenvectors and λ_i are the corresponding eigenvalues of $\mathbf{T}$ by definition we have $\vec{v_i} \cdot \mathbf{T} = \lambda_i \vec{v_i}$
- For $i = 1$ we further know $\lambda_1 = 1$ and thus $\pi \cdot \mathbf{T} = \pi\lambda_1$. With $\pi = \lambda_1 a_i \vec{v_i}$ we arrive at the second-last equation in the slide above.
- We observe that the visitation probabilities at time t can be expressed as the stationary distribution π plus a term that depends on the eigenvalues and the eigenvectors with orders higher than one. The influence of the initial distribution $\pi^{(0)}$ is encoded in the components a_i (which we defined as $\vec{a} := \pi^{(0)}\mathbf{U}^{-1}$).
- Since the transition matrix is stochastic we further know that the largest eigenvalue is one and thus $1 > \lambda_2 \ldots \lambda_n$. For increasing t this implies that the difference between $\pi^{(t)}$ and π vanishes quickly.

Evolution of total variation distance

- for the **total variation distance** $\delta(\pi, \pi^{(t)})$ we have

$$\delta(\pi, \pi^{(t)}) = \frac{1}{2} \sum_{j=1}^{n} |\pi_j - \pi_j^{(t)}| = \frac{1}{2} \sum_{j=1}^{n} \left| \sum_{i=2}^{n} \lambda_i^t a_i (\vec{v_i})_j \right|$$

- for aperiodic and irreducible stochastic matrix $\mathbf{T}$: $\lambda_i < 1$ for $i \geq 2$
- assuming $\lambda_2 > \lambda_3$, for sufficiently large t we can approximate the total variation distance as

$$\delta(\pi, \pi^{(t)}) \approx \frac{1}{2} \sum_{j=1}^{n} |\lambda_2^t a_2 (\vec{v_2})_j|$$

- **how long does it take until visitation probabilities are close to the stationary state?**

Notes:

- In this week's exercise, you have used the total variation distance to quantify how close visitation probabilities at time t are to the stationary distribution. Using the formula from the previous slide, we can actually calculate the total variation distance as a function of time. On the previous slide, we had

$$\pi^{(t)} - \pi = \sum_{i=2}^{n} \lambda_i^t a_i \vec{v_i}$$

- Substituting this in the definition of total variation distance yields an expression for the total variation distance at time t. We can utilise the fact that all eigenvalues are smaller than one.
- How does an increasing value for the (time) exponent t affect the expression of total variation distance? If $\lambda_2 > \lambda_3$, we can always find a t sufficiently large such that the total variation distance is dominated by λ_2 (since for large enough t λ_3^t will be orders of magnitude smaller than λ_2^t)
- With this, we can write a simple approximation for the total variation distance that (i) holds for sufficiently large t, and (ii) depends only on the second-largest eigenvalue of the transition matrix.

Approximating diffusion speed

- for given ϵ we can calculate **minimum time t such that** $\delta(\pi^{(t)}, \pi) \leq \epsilon$ as

$$\delta(\pi, \pi^{(t)}) = \frac{1}{2} \sum_{j=1}^{n} |\lambda_2^t a_2(\vec{v_2})_j| \leq \epsilon \iff$$

$$t \log(|\lambda_2|) - \log(2) + \log\left(\sum_{j=1}^{n} |a_2(\vec{v_2})_j|\right) \leq \log(\epsilon) \iff$$

$$t \geq \frac{1}{\log(|\lambda_2|)} \cdot \left(\log(\epsilon) + \log(2) - \log\left(\sum_{j=1}^{n} |a_2(\vec{v_2})_j|\right)\right) \propto \frac{\log(\epsilon)}{\log(|\lambda_2|)}$$

Notes:

- We can make an approximate analytical prediction of diffusion speed in a given network. In particular, we can calculate how long it takes until visitation probabilities are close to the stationary distribution in terms of total variation distance.
- For a sufficiently small ϵ we can calculate the minimum number of random walk steps t above which the total variation distance is smaller than ϵ.
- For this we start with our expression from the previous slide and take the logarithm on both sides (since we are interested in the exponent t).
- Note that – since $|\lambda_2| < 1$ – we have $\log(|\lambda_2|) < 0$ and we thus must **invert the inequality operator** when dividing by $\log(|\lambda_2|)$. Also note that the absolute value of a product is equal to the product of absolute values of the factors.
- In the resulting expression, the sum $\sum_{j=1}^{n} |a_2(\vec{v_2})_j|$ captures the influence of the initial distribution $\pi^{(0)}$, i.e. if we know the eigenspace transformation of $\pi^{(0)}$ we can compute the actual diffusion speed for any initial distribution.
- However, this is only a constant and for sufficiently large ϵ the term in the brackets will be dominated by $\log(\epsilon)$. We can thus derive a proportionality factor that holds for sufficiently small values of ϵ.
- Our result shows that the time required to achieve a total variation distance smaller than a given ϵ depends on ϵ via a logarithmic function. The dependence on the network topology is captured by the second largest eigenvalue of the transition matrix, which encodes the topology of the network. Both the second-largest eigenvalue and ϵ are smaller than one, hence the negative signs in the counter and the denominator in the expression cancel out.

Practice Session

- we experimentally **study diffusion speed in networks** based on a random walk model
- we compute the time $t(\epsilon)$ after which $\delta(\pi, \pi^{(t)}) < \epsilon$
- we use empirical results to **test our analytical prediction**

practice session

see notebook 12-01 in `gitlab` repository at
→ `https://gitlab.informatik.uni-wuerzburg.de/ml4nets_notebooks/2022_wise_sna_notebooks`

finding

experiments confirm that second-largest eigenvalue λ_2 of transition matrix captures influence of network topology on speed of diffusion process

experimental validation in Watts-Strogatz networks

time $t(\epsilon)$ after which $\delta(\pi, \pi^{(t)}) < \epsilon = 0.005$ (top) and $\frac{\log \epsilon}{\log |\lambda_2|}$ (bottom) for Watts-Strogatz networks with $n = 200$, $s = 5$ and rewiring prob. p (x-axis)

Notes:

- In the first practice session, we experimentally test our analytical prediction for different random microstates frawn from different ensembles. The figures above show the empirical diffusion speed (top) and analytical prediction (bottom) for random networkrs generated by the Watts-Strogatz model.
- We find that the predicted diffusion speed and the empirical diffusion speed coincide almost perfectly. In other words: the second-largest eigenvalue of the transition matrix of a random walk provides a good estimate for the speed of a diffusion process in this network.
- In the resulting expression, we observe that the minimum time increases as we decrease ϵ (since the absolute value of the logarithm of $\epsilon < 1$ increases as ϵ decreases), i.e. we naturally need to wait longer in order to get closer to the stationary distribution.

The spectrum of complex networks

- ordered sequence of eigenvalues $|\lambda_1| > |\lambda_2| > \ldots > |\lambda_n|$ of matrix $\mathbf{T}$ is called **spectrum of T**

spectral gap

We call $1 - |\lambda_2|$ the **eigenvalue gap** or **spectral gap** of $\mathbf{T}$

1. large spectral gap: fast diffusion
2. small spectral gap: slow diffusion

- eigenvalue gap captures **influence of topology on diffusion process**

analytical prediction

$\frac{\log \epsilon}{\log |\lambda_2|}$ for random microstates of one-dimensional Watts-Strogatz model with parameters $n = 200$, $s = 5$ and different rewiring probabilities p (using $\epsilon = 0.005$)

Notes:

- In our experiments, we observe that the smaller the second-largest eigenvalue of the transition matrix, the faster the diffusion speed, i.e. the absolute value of the logarithm of $|\lambda_2|$ increases, which means the predicted time decreases and the speed of diffusion increases.
- Apart from looking at the value of a single eigenvalue λ_2 we can can also consider the full **spectrum of eigenvalues**, i.e. we can consider the list of eigenvalues in ascending order.
- Since the transition matrix is a stochastic matrix in which rows sum to one, we know that the largest eigenvalue is one. Since λ_2 is bounded above by the largest eigenvalue, we can say: the larger the **spectral gap** $1 - |\lambda_2|$, the faster the diffusion speed.
- The spectral gap of the transition matrix can thus be interpreted as a measure that captures the efficiency of the corresponding network with respect to diffusion processes.

Continuous-time diffusion model

model 1

discrete-time Markov chain model for random walks in a network

model 2

continuous-time Markov process model for diffusion of quantities in connected reservoirs

$$\left(\pi_a^{(0)}, \pi_b^{(0)}\right) \cdot \begin{array}{c c} & \begin{array}{cc} a & b \end{array} \\ \begin{array}{c} a \\ b \end{array} & \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \end{array} = \left(\pi_b^{(0)}, \pi_a^{(0)}\right)$$

$$\frac{dx_1(t)}{dt} = C \cdot (x_2(t) - x_1(t))$$

Notes:

- So far, we studied random walks as a model for a diffusion processes, where visitation probabilities of nodes are considered a continuous and conserved quantity that diffuses through the network step by step. This yields a **discrete-time** model for a diffusion process.
- However, this is not the only type of diffusion model that we can think of and in the following we will **study a continuous time model**.
- In the random walk model, the diffusion of visitation probabilities gives rise to a particular type of dynamical system that is goverrned by the repeated multiplication with the transition matrix (shown on the left).
- For the simple example shown above and an initial distribution $(1, 0)$, we observe that this process never reaches a stationary state. This is an artefact of the discrete-time model that we have considered. Importantly this does not match or "physical" intuition of a diffusion process where probabilities represent a conserved, continuous quantity.
- We can instead consider a continuous time process that models the diffusion of physical quantities x_1 and x_2 between two connected reservoirs (figure on the right).
- Apart from the continuous notion of time, in such a physical model we have another important feature: the rate of change of quantities (i.e. the particle flux from one reservoir to the other) depends on the difference between the quantities x_1 and x_2.
- We can model such a diffusion process by means of a continuous differential equation. If the quantities that we describe are probabilities we call this a Kolmogorov equation $\longrightarrow$ Bonus Lecture on Growing Networks

Diffusion in networks vs. diffusion in space

diffusion in network

for diffusion in network with adj. matrix $\mathbf{A}$ we have

$$\frac{dx_i^{(t)}}{dt} = C \cdot \sum_j A_{ij} \left(x_j^{(t)} - x_i^{(t)} \right)$$

$$= C \cdot \sum_j \left(A_{ij} - \delta(i,j) d_i \right) x_j^{(t)}$$

and thus $\frac{dx^{(t)}}{dt} = C \cdot (\mathbf{A} - \mathbf{D}) \cdot x^{(t)}$

where $\mathbf{D}$ is the diagonal degree matrix

diffusion in Euclidean space

for (heat) diffusion in two-dimensional Euclidean space we have

$$\frac{d\psi}{dt} = -C \nabla^2 \psi$$

where ∇^2 is the **Laplacian operator** defined as

$$\nabla^2 \psi(x,y) := \frac{d^2\psi(x,y)}{dx^2} + \frac{d^2\psi(x,y)}{dy^2}$$

Gottfried Wilhelm Leibniz

1646 – 1716

"Space is nothing but a [...] set of **relations among bodies**."

→ G W Leibniz, 1716

image credit: portrait by Christoph Bernhard Francke, Herzog Anton Ulrich-Museum, Braunschweig, public domain

Notes:

- We can generalise the case of two connected reservoirs to a network, by taking into account the adjacency matrix. This is given in the box on the right. The instantaneous rate of change at any given point in time t is described by means of a system of coupled differential equations, where the change is proportional to the difference between quantities x_i and x_j.
- To see how we come to the simplified final equation $C(A-D)x^{(t)}$ in matrix form, we first separate the sums and get $C\sum_j A_{ij}x_j^{(t)} - C\sum_j A_{ij}x_i^{(t)}$. We then use $\sum_j A_{ij} = d(i)$ and get $C\sum_j A_{ij}x_j^{(t)} - Cd(i)x_i^{(t)}$. We can add this to the sum using the the delta-function $\delta(i,j)$ (which is 1 iff $i=j$). We can write this in matrix form by using a **degree diagonal matrix D**, which contains the degrees as diagonal elements.
- We see that the equation in matrix form closely resembles the diffusion equation in a two-dimensional Euclidean space, which uses a differential operator ∇^2 ("nabla") (see box on the right). This resemblance gives us a deep insight into the differences and similarities between continuous Euclidean space and discrete network topologies. In the case of a continuous-time diffusion process on a discrete topology, a matrix takes the role of the so-called Laplacian (differential) operator ∇^2. We can thus see Euclidean space as a continuous two-dimensional structure that captures the relations between objects, just as a network topology captures the relations between nodes. This idea can be traced back to Gottfried Wilhelm Leibniz, who - in a letter to Isaac Newton - argued that space is nothing than a set of relations among bodies (the so-called relational theory of space).

The Laplacian matrix

- defining a **Laplacian matrix** $\mathcal{L} := \mathbf{D} - \mathbf{A}$ we can write

$$\frac{dx^{(t)}}{dt} = -C \cdot \mathcal{L} \cdot x^{(t)}$$

  where $\mathbf{D}$ is the (diagonal) degree matrix with $D_{ii} = d_i$
- Laplacian matrix is **discrete generalization of Laplacian operator to arbitrary network topologies**

undirected network with adjacency matrix $\mathbf{A}$ and degree matrix $\mathbf{D}$

$$\mathcal{L} = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \\ f \\ g \end{array} \begin{array}{c} \begin{array}{ccccccc} a & b & c & d & e & f & g \end{array} \\ \begin{bmatrix} 2 & -1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & -1 & 0 & 0 & 0 \\ -1 & -1 & 2 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 4 & -1 & -1 & -1 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & -1 & -1 & 3 & -1 \\ 0 & 0 & 0 & -1 & 0 & -1 & 2 \end{bmatrix} \end{array}$$

Laplacian matrix $\mathcal{L} = \mathbf{D} - \mathbf{A}$

Notes:

- Reconsidering the equation from the previous slide, we see that we can easily bring it in the form of the diffusion equation in physics.
- We simply change the sign. We find that the (matrix-valued) expression $\mathbf{A} - \mathbf{D}$ takes on the role of the Laplacian operator ∇^2 in the diffusion equation. This is why the matrix $\mathcal{L} = \mathbf{A} - \mathbf{D}$ is called **Laplacian matrix**.
- The Laplacian matrix, which captures the topology of a network, thus provides us with a **discrete generalisation of the Laplacian operator in Euclidean space to arbitrary (discrete) interaction topologies**.
- The Laplacian matrix is just another matrix representation of an (undirected and unweighted) network. Generalizations for weighted and directed cases exist but are unfortunately more complicated to work with $\rightarrow$ F Chung, 2005 .
- The Laplacian matrix is also sometimes called "Kirchhoff matrix". This is due to the fact that the problem of diffusion in networks is intimately related to electrical circuits, which motivated Gustav Kirchhoff to Kirchhoff's matrix tree theorem in which the Laplacian matrix appears naturally.

Diffusion speed and Laplacian eigenvalues

- consider the **matrix ordinary differential equation (ODE)**

$$\frac{dx^{(t)}}{dt} = -C \cdot \mathcal{L} \cdot x^{(t)}$$

- the **standard solution** of such a matrix ODE is

$$x^{(t)} = \sum_{i=1}^{n} \alpha_i e^{-C\lambda_i t} \vec{v_i}$$

where λ_i and $\vec{v_i}$ are i-th eigenvalue and eigenvector of $\mathcal{L}$ respectively

observations

- all eigenvalues $\lambda_i \neq 0 \implies x^{(t)} \to \vec{0}$
- $\lambda_i > 0$ for some $i \implies x^{(t)}$ reaches non-trivial stationary state
- all eigenvalues $\lambda_i = 0 \implies x^{(t)}$ is time-invariant
- larger eigenvalues $\lambda_i \implies$ faster convergence to stationary state

eigenvalues of Laplacian matrix $\mathcal{L}$ determine approach to equilibrium

Notes:

- We can now use the Laplacian matrix to study continuous-time diffusion in a network. Since we have a matrix ordinary differential equation (ODE), we can use the standard general solution based on the exponential function.Remember that a matrix ODE of the form $x'(t) = Ax(t)$ has the general solution $x(t) = \alpha_1 e^{\lambda_1 t} v_1 + \alpha_2 e^{\lambda_2 t} v_2 + \ldots$ if A has n unique eigenvalues.
- This standard solution leads to the following observations:
  1. We need at least one zero eigenvalue to have a non-zero solution, i.e. if all eigenvalues where different from zero, the state $x^(t)$ of the system would converge to the zero vector as $t \to \infty$.
  2. If all eigenvalues $\lambda_i = 0$, then $x^{(t)}$ is time-independent, i.e. the state of the system never changes (since $e^{-C\lambda_i t} = 1$ for all t).
  3. If $\lambda_i > 0$ for some i the system state changes over time but it converges to a stationary state (since $e^{-C\lambda_i t}$ converges to zero for $t \to \infty$).
  4. Larger non-zero eigenvalues λ_i result in a faster convergence (because $e^{-C\lambda_i t}$ converges to zero more quickly and thus we quickly approach the stationary state).
- Here we assume that eigenvector entries sum to zero. Since the Laplacian matrix is symmetric, the eigenvectors form an orthonormal basis, which we can use to express the vector x_i. On slide 3 we used the same idea to express the initial distribution $\pi^{(0)}$ in terms of the eigenvectors of $\mathbf{T}$.

The Laplacian spectrum

- consider eigenvalues $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$ of $\mathcal{L}$ in **ascending order**
- $(1, \ldots, 1) \cdot \mathcal{L} = \vec{0}$ so $\vec{1}$ is eigenvector for **smallest eigenvalue** $\lambda_1 = 0$

example

for example network with two connected components we can arrange $\mathcal{L}$ in two **non-zero block matrices $\mathcal{L}_1$ and $\mathcal{L}_2$** where

1. $\mathcal{L}_1$ and $\mathcal{L}_2$ are Laplacian matrices
2. $\mathcal{L}_1$ and $\mathcal{L}_2$ have eigenvalues 0 (with eigenvectors $(1, 1, 1)$ and $(1, 1, 1, 1)$ respectively)

$$\mathcal{L} = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \\ f \\ g \end{array} \begin{array}{c} \begin{array}{ccccccc} a & b & c & d & e & f & g \end{array} \\ \begin{bmatrix} 2 & -1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 \\ -1 & -1 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & -1 & -1 & -1 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & -1 & -1 & 3 & -1 \\ 0 & 0 & 0 & -1 & 0 & -1 & 2 \end{bmatrix} \end{array}$$

Notes:

- Let us now have a closer look at the eigenvalues of the Laplacian matrix. For this, we order them in **ascending order**. Note that this is different from our study of eigenvalues of the transition matrix $\mathbf{T}$ (where we have used a descending ordering).
- A first observation is due to the special definition of the Laplacian matrix. Since the rows and columns in the (symmetric) Laplacian matrix sum to zero, multiplying an all-one vector $\vec{1} := (1, 1, \ldots, 1)$ with the Laplacian matrix yields the zero vector.
- Since $\vec{1}\mathcal{L} = 0\mathcal{L}$ this implies that the one vector $\vec{1}$ is eigenvector for the (smallest) eigenvalue $\lambda_1 = 0$ of the Laplacian matrix.
- If the network has two connected components, we can reorder rows and columns such that we obtain two block matrices, each by itself being a Laplacian matrix. Each of these matrices has a trivial eigenvalue of zero as well.

Algebraic connectivity

example

for our example network, Laplacian matrix $\mathcal{L}$ has **two zero eigenvalues** $\lambda_1 = \lambda_2 = 0$ with $\vec{v}_1 = (1, 1, 1, 0, 0, 0, 0)$ and $\vec{v}_2 = (0, 0, 0, 1, 1, 1, 1)$

algebraic connectivity

we call the second-smallest eigenvalue λ_2 of the Laplacian matrix **algebraic connectivity**

- $\lambda_2 = 0$ iff network is disconnected
- number of zeros in Laplacian spectrum corresponds to number of connected components → exercise 10

$$\mathcal{L} = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \\ f \\ g \end{array} \begin{array}{c} \begin{array}{ccccccc} a & b & c & d & e & f & g \end{array} \\ \begin{bmatrix} 2 & -1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 \\ -1 & -1 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & -1 & -1 & -1 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & -1 & -1 & 3 & -1 \\ 0 & 0 & 0 & -1 & 0 & -1 & 2 \end{bmatrix} \end{array}$$

$$\lambda_2 = 0$$

Notes:

- For the Laplacian matrix of a network with two connected components, we find that the rows/columns of nodes that belong to each of the connected components individually sum up to zero. This implies that the spectrum of the Laplacian matrix has two zero eigenvalues $0 = \lambda_1 = \lambda_2$ with two different eigenvectors. The one-entries in these eigenvectors correspond to the nodes in the two different connected components respectively.
- This generalises to networks with multiple connected components and the number of zeros in the spectrum of the Laplacian matrix counts the number of connected components. If the network is not connected, i.e. if it has at least two connected components, at least the two smallest eigenvalues λ_1 and λ_2 are 0.
- λ_2 can thus be used to study the connectivity of a network by means of algebraic methods, hence the name: **algebraic connectivity**.
- Think about our comments on algebraic methods in Lecture 02: We have seen that connected components show up as blocks in adjacency matrices (however, for this we needed to assume that rows and columns are suitably ordered). We now found an elegant method to extract this ordering, since we can look at the 0 and 1 components of the eigenvectors corresponding to the zero eigenvalues of the Laplacian matrix.

Algebraic connectivity and diffusion speed

- consider a **connected** network, i.e. $0 < \lambda_2 \leq \lambda_3 \ldots \leq \lambda_n$
- continuous time diffusion equation is

$$x^{(t)} = \sum_{i=1}^{n} \alpha_i e^{-C\lambda_i t} \vec{v_i}$$

- large $\lambda_2 \Rightarrow$ fast diffusion
- algebraic connectivity captures **how well-connected network is**

$$\mathcal{L} = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \\ f \\ g \end{array} \begin{array}{c} \begin{array}{ccccccc} a & b & c & d & e & f & g \end{array} \\ \begin{bmatrix} 2 & -1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & -1 & 0 & 0 & 0 \\ -1 & -1 & 2 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 4 & -1 & -1 & -1 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & -1 & -1 & 3 & -1 \\ 0 & 0 & 0 & -1 & 0 & -1 & 2 \end{bmatrix} \end{array}$$

$\lambda_2 = 0.398$

Notes:

- We can now consider the role of the eigenvalues λ_i in the equation on slide 17.
- If $\lambda_i = 0$ there is no change of $\alpha_i e^{-C\lambda_i t}\vec{v}_i$ as t changes. Since $\lambda_1 = 0$, there is at least one "time-invariant" component in the solution of the differential equation (this gives us a (the) stationary state of the diffusion process). For each additional zero eigenvalue λ_i we get another invariant component, which corresponds to the additional connected components of the network (which have their own independent stationary states).
- If $\lambda_i = 0$ for all i, there is no change at all (since there is no coupling and thus no dynamics). in this case, there are n zeros in the eigenvalue sequence of the Laplacian, meaning that the network consists of isolated nodes, i.e. the initial state is already the stationary state.
- In general, the larger the values of λ_i the faster $x^{(t)}$ will cease to change. Using the same argument as in our discussion of the influence of the second-largest eigenvalue of $\mathbf{T}$, we see that the convergence to the stationary state is actually dominated by the second-smallest eigenvalue $\lambda_2(\mathcal{L})$. From a topology point of view, diffusion speed is negatively influenced by: (i) large diameter; Indeed, algebraic connectivity provides us with an upper bound for diameter (and thus average path length), and (ii) small cuts (i.e. weak ties between communities or cliques); Such small cuts hinder diffusion across communities. Indeed, algebraic connectivity is a lower bound for the size of the smallest cut in a network.
- Using a continuous-time model of a diffusion process, we have found another (spectral) measure for diffusion speed in complex networks.

Practice Session

- we calculate the Laplacian matrix and algebraic connectivity
- we use algebraic connectivity to study connected components
- we compare the eigenvalue gap of transition matrices with algebraic connectivity

practice session

see notebook 12-02 in `gitlab` repository at
→ `https://gitlab.informatik.uni-wuerzburg.de/ml4nets_notebooks/2022_wise_sna_notebooks`

algebraic connectivity

$\frac{\log \epsilon}{\log |\lambda_2|}$ (top) and algebraic connectivity $\lambda_2(\mathcal{L})$ (bottom) for random microstates of one-dimensional Watts-Strogatz model with parameters $n = 200$, $s = 5$ and different rewiring probabilities p

Notes:

- In the practice session, we explore the Laplacian spectrum of networks and study how we can use it to calculate connected components.
- Comparing discrete-time and continuous-time diffusion, the figure above shows the algebraic connectivity (bottom) and the diffusion speed based on the second-largest eigenvalue of the transition matrix (top) for Watts-Strogatz networks with different rewiring probabilities p.
- We find that, even though we have defined it based on a different (continuous-time) diffusion process, the general shape of algebraic connectivity is the same.
- Algebraic connectivity, just like the eigenvalue gap studied based on the transition matrix, captures how well-connected a network topology is.
- One can actually show that algebraic connectivity has important implications for for other types of dynamical processes, such as synchronization, propagation or consensus (see references).

Random walks and communities

- **how do communities influence random walks?**

example

a typical random walk in terms of five community labels A, B, C, D, E of nodes looks like

$$AAAAAAAAABBBBBBBBBBBBBABB$$
$$BBBBBBBBCCCCCCCCCCCCDCCDDD$$
$$DDDDDDDDDDCCCCCCCCCCCBBBBB$$

- we can use random walks to **detect communities** and calculate **node embeddings**
  $\longrightarrow$ Machine Learning for Complex Networks, SoSe 23

- fast diffusion within and slow diffusion across communities is reflected in eigenvectors that capture approach to equilibrium

random walker in network with modularity $Q_{opt} = 0.78$

Notes:

- Above, we argued that the algebraic connectivity captures the presence of small cuts, which limit the speed of diffusion.
- Let us now consider the influence of community structures on a diffusion process (modelled via a random walk process). We observe that the random walker is trapped within communities, switching between communities from time to time.
- This influence of communities on a random walk process, provides an interesting approach to community detection. We can let a random walker explore the network, and use its trajectory to detect communities.
- Some community detection algorithms are based on exactly this influence of community structures on random walks, e.g. the InfoMap algorithm, which we will cover in more detail in our upcoming course *Machine Learning for Complex Networks*. See details in $\longrightarrow$ M Rosvall and CT Bergstrom, 2008
- Similarly, the so-called WalkTrap algorithm uses random walks to detect community structures, see details in $\longrightarrow$ P Pons and M Latapy, 2005
- In the following we will give the idea for a spectral clustering algorithm that utilizes the Laplacian matrix, and that we will implement in practice in tomorrow's practice lecture.

The Fiedler vector

example 1

for example network with two connected components, eigenvector $\vec{v}_2$ corresponding to $\lambda_2 = 0$ is $(1, \ldots, 1, 0, \ldots, 0)$

example 2

for example connected network, eigenvector $\vec{v}_2$ corresponding to $\lambda_2 = 0.398$ is $(-0.49, -0.3, -0.49, 0.21, 0.36, 0.36, 0.36)$

Fiedler vector

- eigenvector $\vec{v}_2$ corresponding to second-smallest Laplacian eigenvalue λ_2 is called **Fiedler vector** → M Fiedler, 1973
- entries of $\vec{v}_2$ can be used **map nodes to connected components or communities** → see Practice Session
- Laplacian spectrum captures influence of topology on evolution of dynamical processes → cf. M Kac, 1966

$$\mathcal{L} = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \\ f \\ g \end{array} \begin{array}{c} \begin{array}{ccccccc} a & b & c & d & e & f & g \end{array} \\ \begin{bmatrix} 2 & -1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & -1 & 0 & 0 & 0 \\ -1 & -1 & 2 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 4 & -1 & -1 & -1 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & -1 & -1 & 3 & -1 \\ 0 & 0 & 0 & -1 & 0 & -1 & 2 \end{bmatrix} \end{array}$$

$$\lambda_2(\mathcal{L}) = 0.398$$

Notes:

- We can use the same idea to detect communities based on the Laplacian spectrum of a network.
- For a network with two disconnected components, we have seen that zero- and one-entries of the eigenvector corresponding to the algebraic connectivity can be used to map nodes to connected components
- Interestingly, this result translates to general networks with a single connected component as well. Here, the *sign* of entries in the eigenvector corresponding to the algebraic connectivity can be used to map nodes to communities.
- This eigenvector is called **Fiedler vector** in honour of the Czech mathematician Miroslav Fiedler, who pioneered the use of the eigenvalues of Laplacian matrices of graphs in his seminal work "Algebraic Connectivity of Graphs" (1973).
- Since the sign of the entries can necessarily only bisect the network, we also speak of "spectral bisection" of networks. This approach can be applied recursively to detect more than two communities. We will explore this in an example in exercise 10.
- This brings us back to our discussion from lecture L02, where we have seen that communities show up as blocks of high values in the powers of (properly sorted) adjacency matrices. Back then, we missed a way to find the right ordering of rows/columns such that blocks of high/low values correspond to communities. We now understand that we can use the entries of eigenvectors to find this right "ordering" of rows/columns, thus giving rise to a spectral clustering algorithm for networks.

Practice Session

- we compute the **Fiedler vector** of a network
- we use the Fiedler vector and the algebraic connectivity to **detect community structures in networks**

practice session

see notebook 12-03 in `gitlab` repository at
→ `https://gitlab.informatik.uni-wuerzburg.de/ml4nets_notebooks/2022_wise_sna_notebooks`

12-03: Spectral Clustering

January 26 2022

In a final practice session, we use our knowledge about Laplacian spectra to detect community structures in networks. We demonstrate the resulting spectral partitioning algorithm in a small synthetic example.

```python
import pathpy as pp
import numpy as np
from tqdm.notebook import tqdm

import seaborn as sns
import matplotlib.pyplot as plt

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

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

✓ 0.4s Python

```python
n = pp.Network(directed=False)
n.add_edge('a', 'b')
n.add_edge('b', 'c')
n.add_edge('a', 'c')
n.add_edge('b', 'd')
n.add_edge('d', 'f')
n.add_edge('d', 'g')
n.add_edge('d', 'e')
n.add_edge('e', 'f')
n.add_edge('f', 'g')
n
```

✓ 0.4s Python

```
<pathpy.models.network.Network object at 0x000001CA0402B4F0>
```

The Fiedler vector

Let us write a function `fiedler` that computes the so-called **Fiedler vector** of a given undirected and unweighted network, i.e. the eigenvector corresponding to the algebraic connectivit of the Laplacian matrix, as defined in Lecture 09.

```python
def laplacian(network):
    A = network.adjacency_matrix()
    D = sp.sparse.diags(pp.statistics.degree_sequence(network))
    L = D - A
    return L

def algconn(network):
    L = laplacian(network)
    # "SM" stands for smallest magnitude, k=2 yields the first two eigenvalues in the sorted sequence
    w = sp.sparse.linalg.eigs(L, which="SM", k=2, return_eigenvectors=False)
    eigenvalues_sorted = np.sort(np.absolute(w))
    return eigenvalues_sorted[1]
```

Notes:

- In the third practice session we use our theoretical insights to implement a function that uses the Laplacian spectrum of a network to detect community structures.

Matrix representations of networks

adjacency matrix

- simplest matrix representation
- rows/columns sum to in/out-degrees
- basis for definition of transition matrix and Laplacian matrix
- largest eigenvector: eigenvector centrality
- directed, undirected and weighted networks

transition matrix

- random walk transition probabilities
- stochastic matrix, i.e. rows sum to 1
- largest eigenvalue 1
- largest eigenvector: stationary distribution
- second-largest eigenvalue: diffusion speed
- directed, undirected and weighted networks

Laplacian matrix

- generalisation of Laplacian operator to networks
- rows and columns sum to 0
- smallest eigenvalue 0
- second-smallest eigenvalue: algebraic connectivity
- second-smallest eigenvector: Fiedler vector
- undirected networks (for directed generalization see → F Chung, 2005)

$$\mathbf{A} = \begin{pmatrix} 0&1&1&0&0&0&0 \\ 1&0&1&1&0&0&0 \\ 1&1&0&0&0&0&0 \\ 0&1&0&0&1&1&1 \\ 0&0&0&1&0&1&0 \\ 0&0&0&1&1&0&1 \\ 0&0&0&1&0&1&0 \end{pmatrix}$$

$$\mathbf{T} = \begin{pmatrix} 0&\frac{1}{2}&\frac{1}{2}&0&0&0&0 \\ \frac{1}{3}&0&\frac{1}{3}&\frac{1}{3}&0&0&0 \\ \frac{1}{2}&\frac{1}{2}&0&0&0&0&0 \\ 0&\frac{1}{4}&0&0&\frac{1}{4}&\frac{1}{4}&\frac{1}{4} \\ 0&0&0&\frac{1}{2}&0&\frac{1}{2}&0 \\ 0&0&0&\frac{1}{3}&\frac{1}{3}&0&\frac{1}{3} \\ 0&0&0&\frac{1}{2}&0&\frac{1}{2}&0 \end{pmatrix}$$

$$\mathcal{L} = \begin{pmatrix} 2&-1&-1&0&0&0&0 \\ -1&3&-1&-1&0&0&0 \\ -1&-1&2&0&0&0&0 \\ 0&-1&0&4&-1&-1&-1 \\ 0&0&0&-1&2&-1&0 \\ 0&0&0&-1&-1&3&-1 \\ 0&0&0&-1&0&-1&2 \end{pmatrix}$$

Notes:

- In summary, we have studied networks based on three different matrix representations. Let us summarise their properties.
    1. The **adjacency matrix** is the simplest representation of a network and its leading eigenvector can be used to compute the eigenvector centrality of nodes.
    2. The **transition matrix** is a stochastic matrix, i.e. its rows sum to 1. Its largest eigenvalue is 1 and this eigenvalue is unique if P is irreducible and aperiodic (note that this results from the Peron-Frobenius theorem). The magnitude of the second-smallest eigenvalue of the transition matrix can be used to calculate the convergence speed of a random walk process.
    3. The **Laplacian matrix** can be used to analyze a continuous-time diffusion process. Due to its special construction, rows and columns sum to 0. This implies that its smallest eigenvalue is always 0. The number of zeros in the eigenvalue sequence corresponds to the connected components. The second-smallest eigenvalue is the algebraic connectivity and the corresponding eigenvector is the Fiedler vector.

In summary

- we can predict diffusion speed in a network based on **eigenvalue spectrum** of matrix representations

related question

how many times do we need to shuffle a deck of cards before we have "forgotten" the initial configuration? → S Assaf, 2011

- eigenvalues and eigenvectors of matrix representations are basis for **spectral network analysis**

related question

can we make statements about the shape of a drumhead based on the sound of the drum?

→ M Kac, 1966

two (isospectral) drumheads with different shapes but identical harmonics

image credit: Jitse Niesen, Wikimedia Commons, public domain

Notes:

- In summary, we used the eigenvalues of matrix representations to analytically study diffusion processes, namely the speed at which the process converges to the stationary state. Starting in an arbitrary initial state, over time we forget the initial state and approach a unique stationary state. Hence, each step of a random walk destroys information about the initial distribution. This loss of information has an intriguing relation to a phenomenon you are familiar with: we shuffle cards to destroy the initial order in the card deck. We need to shuffle sufficiently long until the deck has "forgotten" the initial order (i.e. distribution) of cards. The question how long one needs to shuffle to forget this initial state (and approach a stationary distribution) can be actually be addressed mathematically using random walks and spectral graph theory. For a deck of 52 cards, you need to shuffle at least 7 times to reach a total variation distance smaller than 0.5 and it decreases by a factor of two with each additional shuffle (depending on the shuffling). → A Assaf et al., 2011
- We further used eigenvalues and eigenvectors to learn something about the network. With this we effectively observe a process, trying to learn something about the network in which it evolves. This has interesting relations to another problem: Consider you hear the sound of a drum. Can you say something about the shape of the drumhead? This is effectively the same problem, because the harmonics that govern the sound of the drum is determined by the eigenvalues and eigenvectors of the underlying drum topology. It has been shown that we cannot actually determine the precise shape of the drumhead, because different drumheads can have the same spectrum, see → M Kac, 1966 . Similarly it has been shown that different (so-called isospectral) networks can have the spectrum. → L Collatz and U Sinogowitz, 1957

Exercise sheet 10

- **last exercise sheet** is available on WueCampus
  - deepen your understanding of Laplacian eigenvalues and spectral clustering
  - use graph Laplacians to addressing visualization and embedding tasks
- solutions are due **February 1st** (via WueCampus)
- present your solution to earn bonus points

Statistical Network Analysis
WiSe 2021/2022

Prof. Dr. Ingo Scholtes
Chair of Informatics XV
University of Würzburg

Exercise Sheet 10

Published: January 25, 2022
Due: February 2, 2022
Total points: 10

This is the final exercise sheet for this semester

1. **Algebraic Connectivity and Spectral Clustering**

   (a) Show that the eigenvalue sequence of a Laplacian matrix $\mathcal{L}(G)$ of a network G with k connected components has k zeros. 2P

   (c) Consider a simple model for networks consisting of two equally-sized clusters, where each cluster is a random Erdös-Renyi network and the two clusters are connected by "crossing" a configurable number of random links drawn from both clusters. We have implemented this model in a practice session accompanying lecture 12. Study for which value of the randomly crossed link pairs the cluster structures can be reliably detected based on the simple spectral clustering algorithm implemented in the practice session. 3P

2. **Line Graphs and Laplacian Embedding**

   (a) Arc diagrams are a simple method to visualize networks. In an arc diagram, we places the nodes on a (virtual) horizontal line and connect them with semicircles that can be drawn above or below the line. Like most network visualizations, arc diagrams are more intelligible when there are fewer crossings between the lines that depict edges. A (somewhat naive) strategy to minimize crossings is to place nodes that are connected by an edge closer to each other. The cumulative distance between connected nodes can be quantified by the sum of squared distances of all node pairs connected by an edge, which we denote as Δ^2. 3P

   We can now consider visualization algorithms that minimize this distance. Without further constraints the minimization of Δ^2 is subject to trivial solutions: First, we can place all nodes at the same coordinate, which trivially minimizes Δ^2. Similarly, we can trivially improve any assignment of nodes to (one-dimensional) coordinates $\mathbf{x}$ by simply rescaling them to $\alpha\mathbf{x}$ ($\alpha < 1$). To avoid this we thus impose the constraints that (i) the coordinates x_i of nodes i cannot be the same, and (ii) the assignment $\mathbf{x}$ has a fixed scale, e.g. $\|\mathbf{x}\| = 1$.

   Given a network with adjacency matrix $\mathbf{A}$ show that, subject to the constraints above, the vertex positions that minimize Δ^2 are given by $\arg\min_{\mathbf{x}\in\{\mathbf{x}:\mathbf{x}\perp\mathbf{1}\,\wedge\,\|\mathbf{x}\|=1\}} \mathbf{x}^{\mathbf{T}}\mathcal{L}\mathbf{x}$ where $\mathcal{L}$ is the Laplacian matrix.

   (b) The method explained in the previous task generates a so-called "embedding" of a network in a one-dimensional space. This method is not limited to embedding networks on a line. It can also be used to embed a network in a space of any dimensionality $0 < d \leq rank(\mathcal{L})$. Use the eigenvectors of the two smallest non-zero eigenvalues of the Laplacian matrix as coordinates to embed nodes in the following network topologies in a two-dimensional Euclidean space: 2P

   - a two-dimensional grid lattice network
   - a three-dimensional cube lattice network
   - an Erős-Rényi network
   - a small empirical network from the `konect` database

   Compare these visualizations with the ones returned by `pathpy`'s force-directed layout. Discuss the performance of the eigenvector-based strategy in the empirical network.

Notes:

Self-study questions

1. What are the largest eigenvalues of a transition matrix, and adjacency matrix, and a Laplacian matrix.
2. What can we learn from the eigenvalue spectrum of the transition matrix?
3. What is the spectral gap? How can we use it to predict diffusion speed??
4. What is the minimum time t above which the total variation distance of a random walk with transition matrix $\mathbf{T}$ falls below a given ϵ?
5. How is the algebraic connectivity of a network defined and what does its magnitude tell us about the network topology.
6. How can we use the Laplacian matrix to calculate the number of connected components of an undirected network?
7. What is the algebraic connectivity of a fully connected, undirected network with n nodes?
8. Give an example for two undirected networks that have the same Laplacian eigenvalue spectrum.
9. What is the Fiedler vector of a network? What does it tell us about the topology?

Notes:

References

reading list

- N Masuda ansd R Lambiotte: **Random walks and diffusion on networks**, Physics Reports, Volumes 716–717, 2017
- S Assaf, P Diaconis, K Soundararajan: **A rule of thumb for riffle shuffling**, The Annals of Applied Probability, 2011
- EH Moore: **On the reciprocal of the general algebraic matrix**, Bulletin of the American Mathematical Society, 1920
- R Penrose: **On best approximate solution of linear matrix equations**, Proceedings of the Cambridge Philosophical Society, 1956
- L Collatz, U Sinogowitz: **Spektren endlicher Grafen**, Abh. Math. Sem. Univ. Hamburg, 1957
- M Kac: **Can One Hear the Shape of a Drum?**, American Mathematical Monthly, 1966
- M Fiedler: **Algebraic connectivity of graphs**, Czechoslovak Math. J., 1973
- P Pons and M Latapy: **Computing Communities in Large Networks Using Random Walks**, Computer and Information Sciences ISCIS, 2005
- F Chung: **Laplacians and the Cheeger Inequality for Directed Graphs**, Annals of Combinatorics, 2005
- M Rosvall and CT Bergstrom: **Maps of information flow reveal community structure in complex networks**, PNAS, 2008

A RULE OF THUMB FOR RIFFLE SHUFFLING

SAMI ASSAF, PERSI DIACONIS, AND K. SOUNDARARAJAN

ABSTRACT. We study how many riffle shuffles are required to mix n cards if only certain features of the deck are of interest, e.g. suits disregarded or only the colors of interest. For these features, the number of shuffles drops from $\frac{3}{2}\log_2 n$ to $\log_2 n$. We derive closed formulae and an asymptotic 'rule of thumb' formula which is remarkably accurate.

1. INTRODUCTION

In this paper we study the mixing properties of the Gilbert-Shannon-Reeds model for riffle shuffling n cards. Informally, the deck is cut into two piles by the binomial distribution, and the cards are riffled together according to the rule: if the left packet has A cards and the right has B cards, drop the next card from the left packet with probability $A/(A+B)$ (and from the right packet with probability $B/(A+B)$). Continue until all cards have been dropped. This defines a measure, denoted $Q_2(\sigma)$, on the symmetric group $\mathcal{S}_n$. Repeated shuffles are defined by *convolution powers*

$$Q_2^{*k}(\sigma) = \sum_{\tau \in \mathcal{S}_n} Q_2(\tau) Q_2^{*(k-1)}(\sigma\tau^{-1}). \tag{1}$$

The *uniform distribution* is $U(\sigma) = 1/n!$. There are several notions of the distance between Q_2^k and U: the *total variation distance*

$$\|Q_2^k - U\|_{TV} = \max_{A \subset \mathcal{S}_n} |Q_2^{*k}(A) - U(A)| = \frac{1}{2} \sum_{\sigma \in \mathcal{S}_n} |Q_2^{*k}(\sigma) - U(\sigma)|, \tag{2}$$

and the *separation* and l_∞ metrics

$$\mathrm{SEP}(k) = \max_\sigma 1 - \frac{Q_2^{*k}(\sigma)}{U(\sigma)}, \qquad l_\infty(k) = \max_\sigma \left| 1 - \frac{Q_2^{*k}(\sigma)}{U(\sigma)} \right|. \tag{3}$$

In widely cited works, Aldous [2] and Bayer and Diaconis [5] show that $\frac{3}{2}\log_2(n) + c$ shuffles are necessary and sufficient to make the total variation distance small, while $2\log_2(n) + c$ shuffles are necessary and sufficient to make separation and l_∞ small.

The distances in (2) and (3) look at all aspects of a permutation. In many card games, only some aspects of the permutation matter. For example, in Black-Jack and Baccarat, suits are irrelevant and all 10's and picture cards are equivalent; ESP card guessing experiments use a Zener deck of 25 cards with each of 5 symbols repeated five times. It is natural to ask how many shuffles are required in these situations. These questions are studied by

Date: November 7, 2018.
2000 *Mathematics Subject Classification.* Primary 60B15; Secondary 60C05.
Key words and phrases. card shuffling, cutoff phenomenon, lumping of Markov chains, Poisson summation.

1

Notes: