Lecture 10: Graphs, Networks, and Markov Dynamics

Modelling Complex Systems (Spring 2026)

Jordi-Lluís Figueras

Matematiska Institutionen, Uppsala Universitet

April 27, 2026

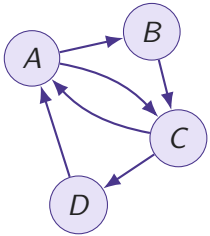
Prelecture discussion

Before Lecture 10, think about a familiar city map as a system of connected places.

- Which locations would you choose as the nodes: neighborhoods, metro stations, or intersections?
- Which connections would you choose as the edges: roads, walking routes, or commuting flows?
- If people move every day from one neighborhood to another, what data would you need to model that movement?

Questions to reflect on:

- When is a network naturally *directed*?
- What makes a transition rule probabilistic rather than deterministic?
- What long-time behavior would you expect if movement patterns are repeated every day?



city neighborhoods as a network

Lecture goals

In this lecture we will:

- define the basic notions of a graph and a network;
- explain why networks are natural models for transport, communication, and interaction systems;
- introduce Markov chains as dynamical systems on graphs;
- formulate the city-population example in matrix form;
- interpret the evolution as a linear dynamical system;
- explain which spectral questions control the long-time behavior;
- state the Perron-Frobenius theorem and connect it to stationary distributions.

Complex Networks

The third module of the course

This lecture begins the third module of the course: complex networks.

In this module we move from low-dimensional deterministic systems and cellular automata to systems whose behavior is shaped by a network of interactions.

The central modelling question is:

Guiding idea

How does the structure of the underlying graph influence the dynamics that takes place on it?

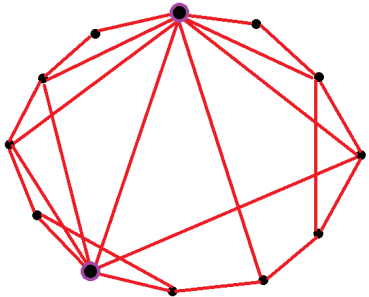
We begin with examples, then introduce graphs, networks, and Markov dynamics.

Application: small-world structure

Small-world networks are one of the classical reasons graphs became central in complex-systems modelling.

- social and communication networks often have strong local clustering;
- a small number of long-range links can make typical graph distances very short;
- this helps explain why information, influence, or disruptions can spread quickly.

The key message is that topology alone can strongly affect the dynamics.



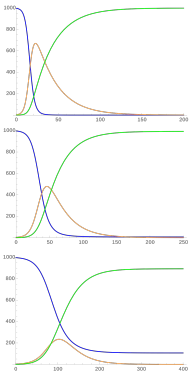
Source: Wikimedia Commons, `File:Small-world-network-example.png` (Schulllz, CC BY-SA 3.0).

Application: epidemics on contact networks

In epidemic models, nodes can represent people, households, or cities, while edges represent possible transmission paths.

- network heterogeneity changes who infects whom;
- highly connected nodes may accelerate the outbreak;
- mobility or contact structure affects both the peak and the final size of the epidemic.

This is one reason graph-based SIR and metapopulation models are so useful.



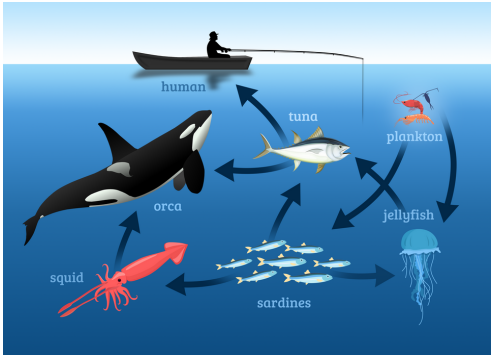
Source: Wikimedia Commons, `File:SIR_flattening_the_curve_3variants.svg` (Phrontis, CC BY-SA 3.0).

Application: population dynamics on ecological networks

Population dynamics also appears naturally on graphs.

- nodes may represent habitat patches, species, or trophic groups;
- edges may represent migration, dispersal, predation, or resource flows;
- the network structure helps determine persistence, recolonization, and cascade effects.

This gives a bridge from graph theory to ecology and spatial population models.



Source: Wikimedia Commons,

`File:Marine_food_web_example.png` (MAKY.OREL, CC0).

Graphs and Networks

Why a new modelling block?

Many complex systems are not naturally organized by position on a line or a grid.

Instead, what matters is a pattern of connections:

- people connected by friendship or communication links;
- computers connected by data-routing links;
- airports connected by flights;
- neighborhoods connected by commuting flows.

This leads to a new abstraction:

Central idea

The state space is organized by a graph, and the dynamics is constrained by the edges of that graph.

What is a graph?

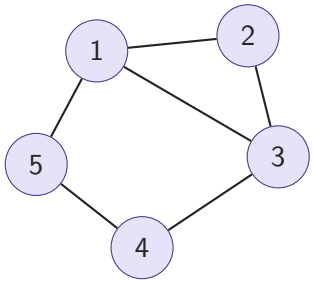
A graph is a pair $G = (V, E)$ where:

- V is a set of vertices or nodes;
- E is a set of edges describing which pairs of vertices are connected.

Two common versions are:

- **undirected graph:** an edge $\{i, j\}$ means that i and j are linked symmetrically;
- **directed graph:** an edge (i, j) means that one can move or send influence from i to j.

The graph records the *topology of allowed interactions*, not yet the strength or meaning of those interactions.



a graph: nodes and edges

Graph versus network

In practice, the word network usually means more than a bare graph.

Typical extra structure includes:

- edge weights,
- directions,
- capacities,
- probabilities,
- node attributes.

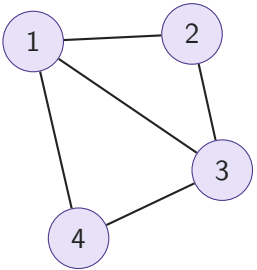
So a useful working distinction is:

Working distinction

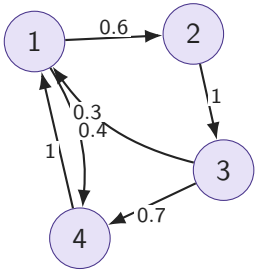
A graph tells us *who is connected to whom*. A network adds quantitative or functional information to those connections.

For modelling, this extra structure is usually what turns a combinatorial object into a dynamical system.

A first visual example



undirected graph



directed weighted network

The same set of nodes can support very different models depending on what extra information is attached to the edges.

How graphs become matrices

For a graph with n nodes, we often encode connections in a matrix.

The simplest choice is the adjacency matrix $A \in \mathbb{R}^{n \times n}$:

$$A_{ij} = \begin{cases} 1 & \text{if there is an edge from } i \text{ to } j, \\ 0 & \text{otherwise.} \end{cases}$$

For weighted networks, A_{ij} can store the weight instead of only 0 or 1.

This is important because it turns a network question into a linear algebra question.

From static structure to dynamics

A graph by itself is static. To get a dynamical system, we need a rule for how some quantity changes over time.

Examples of node-based quantities are:

- population at each neighborhood,
- probability of a random walker being at each node,
- amount of traffic at each junction,
- level of activation in each part of a network.

Today we begin with the cleanest case: a population that redistributes across the nodes according to fixed transition probabilities.

Markov Chains on Graphs

Why do we need stochastic processes?

So far in the course, many of our dynamical systems were deterministic: once the initial condition is fixed, the future is completely determined.

But many systems on networks do not work like that.

If we follow one person, one packet of information, or one random walker, there may be several possible next moves, each with some probability.

Informally, a stochastic process is just a time-evolving system with randomness in it. We need this language because the next state is not given by one fixed rule alone: it is given by a *distribution of possible outcomes*.

So this is not like the deterministic dynamical systems from before. Here, randomness is part of the model itself.

What is a Markov chain?

A Markov chain is a stochastic process that jumps between finitely many states.

The defining property is the memoryless property:

Markov property

The distribution of the next state depends only on the present state, not on the whole past history.

If the states are the nodes of a graph, then a jump is allowed only along edges of that graph. So a Markov chain can be viewed as a dynamical system constrained by a network geometry.

Transition probabilities

Suppose the nodes are $1, \ldots, n$. For each pair of nodes, let P_{ij} be the probability of moving from node j to node i in one time step.

Then:

- $P_{ij} \geq 0$ for all i, j;
- for each fixed node j, the probabilities of all possible destinations must add up to 1:

$$\sum_{i=1}^{n} P_{ij} = 1.$$

So P is a column-stochastic matrix. Zeros in P encode forbidden moves, which come from missing directed edges in the graph.

City example: four neighborhoods

Consider a city with four neighborhoods:

$$\text{Downtown } (D), \text{ University } (U), \text{ Residential } (R), \text{ Industrial } (I).$$

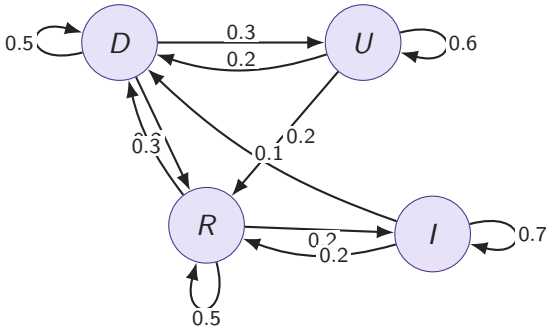
Each evening, a person currently in one neighborhood chooses where to be the next morning according to fixed probabilities.

For example:

- from Downtown, 50% stay in D, 30% go to U, 20% go to R;
- from University, 60% stay in U, 20% go to D, 20% go to R;
- from Residential, 50% stay in R, 30% go to D, 20% go to I;
- from Industrial, 70% stay in I, 20% go to R, 10% go to D.

This already defines a directed weighted network.

City example as a graph



This picture is the network version of the verbal description: every outgoing edge from one neighborhood carries the probability of the next move.

The city as a transition matrix

With the node order (D, U, R, I), the transition matrix is

$$P = \begin{pmatrix} 0.5 & 0.2 & 0.3 & 0.1 \\ 0.3 & 0.6 & 0 & 0 \\ 0.2 & 0.2 & 0.5 & 0.2 \\ 0 & 0 & 0.2 & 0.7 \end{pmatrix}.$$

Each column describes where the population currently at one neighborhood goes next.

Check:

$$0.5 + 0.3 + 0.2 + 0 = 1, \qquad 0.2 + 0.6 + 0.2 + 0 = 1,$$

and similarly for the other columns.

There is also a Python script for this example

To make the city example more concrete, there is a small script:

`scripts/cityMarkovChain.py`

It lets you:

- iterate the Markov chain for several time steps;
- change the initial population vector;
- compare the evolving population with the stationary distribution.

So if you want to see the matrix dynamics in action, this script is the computational version of the example from the slides.

Population evolution

Let

$$x_t = \begin{pmatrix} x_{D,t} \\ x_{U,t} \\ x_{R,t} \\ x_{I,t} \end{pmatrix}$$

denote the population vector at time t.

Then the Markov update rule is simply

$$x_{t+1} = P x_t.$$

So the population after two days is $x_{t+2} = P^2 x_t$, and after k days it is

$$x_{t+k} = P^k x_t.$$

This is why the long-time dynamics is controlled by the powers of the matrix P.

Why total population is preserved

Because each column of P sums to 1, the total population stays constant:

$$\sum_{i=1}^{n}(x_{t+1})_i = \sum_{i=1}^{n}\sum_{j=1}^{n} P_{ij}(x_t)_j = \sum_{j=1}^{n}\left(\sum_{i=1}^{n} P_{ij}\right)(x_t)_j = \sum_{j=1}^{n}(x_t)_j.$$

So the dynamics redistributes population across the graph, but does not create or destroy total mass.

If we normalize so that the total population is 1, then x_t is a probability distribution over the nodes.

One numerical step in the city example

Suppose initially

$$x_0 = \begin{pmatrix} 4000 \\ 2000 \\ 3000 \\ 1000 \end{pmatrix}.$$

Then

$$x_1 = Px_0 = \begin{pmatrix} 3400 \\ 2400 \\ 2900 \\ 1300 \end{pmatrix}.$$

Interpretation:

- Downtown loses some people to other neighborhoods;
- University gains population from Downtown;
- Industrial grows through the Residential–Industrial connection.

Each additional day is obtained by multiplying once more by the same matrix.

Markov chains as linear dynamical systems

From the modelling viewpoint, this is a major simplification.

We have a discrete-time system of the form

$$x_{t+1} = Px_t,$$

which is a linear dynamical system.

That means we can ask familiar questions:

- Does the system converge?
- Is there an equilibrium distribution?
- Is the limit independent of the initial condition?
- How fast is convergence?

To answer these, we must understand the spectrum and eigenvectors of P.

How Do We Study the Dynamics?

Fixed points and stationary distributions

An equilibrium for the update $x_{t+1} = Px_t$ is a vector x^* such that

$$Px^* = x^*.$$

So a stationary state is exactly an eigenvector with eigenvalue 1.

If x^* has nonnegative entries summing to 1, then it is called a stationary distribution.

This immediately connects the probabilistic model with linear algebra:

Key translation

Long-time behavior of the Markov chain is governed by the eigenvalues and eigenvectors of the transition matrix.

Why eigenvalues matter

Very roughly, powers of a matrix are controlled by its eigenvalues.

If all eigenvalues other than 1 have modulus strictly smaller than 1, then repeated multiplication by P damps out those components.

So one expects

$$P^k x_0 \longrightarrow x^* \qquad \text{as } k \to \infty,$$

where x^* is a stationary distribution.

This is the basic mechanism behind convergence to equilibrium in many network-transport models.

Possible obstructions

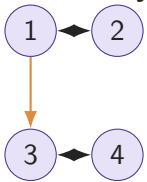
Convergence is not automatic for every stochastic matrix. Typical obstructions are:

- **reducibility:** the graph splits into disconnected communicating parts;
- **periodicity:** the dynamics cycles between classes of states;
- **multiple closed classes:** different long-time outcomes depend on the initial condition.

So we need structural assumptions on the graph and positivity assumptions on the matrix. Those assumptions are precisely where Perron-Frobenius enters.

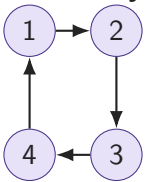
Possible obstructions: visual

Reducibility



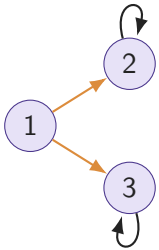
Two communicating blocks are present, but there is no return from the lower block to the upper one.

Periodicity



A pure directed cycle forces the walk to return only at multiples of the cycle length.

Multiple closed classes



The long-time outcome depends on which closed class eventually captures the mass.

Perron-Frobenius theorem: informal message

The Perron-Frobenius theorem applies to matrices with nonnegative entries. It tells us that positivity forces a strong spectral structure.

For Markov chains, the practical message is:

- there is a distinguished dominant eigenvalue;
- it has a nonnegative eigenvector;
- under stronger positivity assumptions, that dominant direction is unique and attracts the dynamics.

This is exactly what we need to justify convergence to a unique long-time population profile.

Perron-Frobenius theorem

Perron-Frobenius theorem for a positive matrix

Let $A \in \mathbb{R}^{n \times n}$ have strictly positive entries: $A_{ij} > 0$ for all i, j. Then:

1. A has a real eigenvalue $\rho(A) > 0$ equal to its spectral radius;
2. there is an eigenvector v associated with $\rho(A)$ whose entries are all strictly positive;
3. the eigenvalue $\rho(A)$ is simple;
4. every other eigenvalue λ satisfies $|\lambda| < \rho(A)$.

There are more general versions for irreducible or primitive nonnegative matrices, which are the forms most relevant to Markov chains.

What the theorem says for stochastic matrices

For a stochastic matrix P, the special eigenvalue is already known:

$$\rho(P) = 1.$$

Indeed, total mass is preserved, so 1 is the natural scale of the dynamics.

If P is sufficiently connected and aperiodic, Perron-Frobenius implies:

- there exists a unique stationary distribution x^*;
- x^* has strictly positive entries;
- for every initial distribution x_0,

$$P^k x_0 \to x^*.$$

So the dynamics forgets the initial condition and settles to a network-dependent equilibrium.

Back to the city interpretation

In the city example, a stationary distribution means:

after one more day of movement, the expected population in each neighborhood is unchanged.

This does *not* mean that individuals stop moving. It means that the aggregate inflow and outflow balance at each node.

That is a recurring theme in complex systems:

Microscopic motion, macroscopic stationarity

Agents may keep moving all the time, while the large-scale distribution becomes stable.

What to remember

- A graph records possible connections; a network enriches those connections with quantitative meaning.
- A Markov chain is a stochastic dynamics on the nodes of a graph.
- With a population vector x_t, the evolution is the linear update $x_{t+1} = Px_t$.
- Long-time behavior is therefore a question about powers of a nonnegative matrix.
- Stationary distributions are eigenvectors for eigenvalue 1.
- Perron-Frobenius explains why positivity and connectivity often produce a unique dominant equilibrium profile.

This will be the starting point for more advanced network dynamics later on.

Acknowledgment

These slides were prepared with the assistance of OpenAI Codex / ChatGPT 5.4 as a drafting and editing tool.
All mathematical, pedagogical, and course-specific content remains under the responsibility of the lecturer.