Concentration effects and collective variables in dynamical systems on networks

Dissertation

zur Erlangung des Grades eines

Doktors der Naturwissenschaften
- Dr. rer. nat -

am Fachbereich Mathematik und Informatik
der Freien Universität Berlin

vorgelegt von

Marvin Lücke

Berlin, 2024

Betreuer: Prof. Dr. Péter Koltai

Erstgutacher: Prof. Dr. Péter Koltai
Zweitgutacher: Prof. Dr. Christian Kühn

Tag der Disputation: 30.10.2024

Acknowledgments

First and foremost, I would like to thank Péter Koltai, Stefanie Winkelmann, Nora Molkenthin, and Jobst Heitzig for giving me the opportunity to work on this topic. I am grateful for your support and guidance throughout the last years and for the numerous interesting discussions we had. A special thanks goes to Péter Koltai for being my PhD supervisor and to Stefanie Winkelmann for welcoming me into her group at the Zuse Institute Berlin.

I would also like to thank all of my other colleagues at the Zuse Institute Berlin, Freie Universität Berlin, and Potsdam Institute for Climate Impact Research for the insightful chats and helpful advice that they provided. In particular, I express my gratitude to all those who assisted me in improving this work.

This thesis has been supported by the Deutsche Forschungsgemeinschaft (DFG) under Germany's Excellence Strategy via the Berlin Mathematics Research Center MATH+ (EXC2046/ project ID: 390685689).

Abstract

Dynamical systems on networks, which can be seen as a special type of *agent-based model*, are widely used to model systems that consist of many interacting entities called *agents*. In this framework the nodes in the network represent the agents, the edges represent the relations between them, and the *state* of each agent evolves over time in dependence on the states of its neighbors, typically in a stochastic manner. Although the state evolution of each single agent is often dictated by simple rules and mechanisms, the overall *collective* or *emergent* behavior of the system, which is the result of many individual interactions, can be incredibly hard to anticipate and understand. The investigation of this collective behavior is the main focus of this thesis.

Even though the collective behavior is difficult to predict, it is typically much less complex than the vast number of degrees of freedom would allow for and instead (approximately) follows some low-dimensional dynamics. The understanding of collective behavior hence consists of two steps. Firstly, a projection that maps the high-dimensional *microscopic state*, containing the state of each agent, to a low-dimensional *macroscopic state*, containing only the essential aggregated information to describe the collective behavior, has to be found. This projection, which filters out unnecessary degrees of freedom and quickly decaying processes of the original system, is called a *collective variable* (CV). Secondly, the reduced macroscopic system that dictates the evolution of the macroscopic state has to be derived. If the choice of CV was appropriate, this macroscopic system is able to reproduce the low-dimensional projection of the original model, i.e., the collective behavior. Akin to the law of large numbers, the aggregated random actions of many agents sometimes lead to an approximately deterministic macroscopic dynamics, which is referred to as a *concentration effect*.

This thesis mainly considers Markovian discrete-state dynamics on (random) networks and addresses efficient simulation, discovery of CVs and macroscopic dynamics, and the occurrence of concentration effects. In this setting the *shares* of each discrete state in the system, or in certain subsystems, constitute an important choice of CVs. Conditions that guarantee the convergence of the dynamics of the shares to a deterministic mean-field ordinary differential equation in the large population limit are proved. These conditions enable the derivation of parameter bounds for popular random graph models, e.g., Erdős–Rényi random graphs, the stochastic block model, and random regular graphs, that ensure convergence to the mean-field limit, which is demonstrated for a continuous-time noisy voter model (CNVM). For systems that do not exhibit convergence to the mean-field limit and for which the simple state shares are not appropriate CVs because they lack essential state information, a data-driven method for algorithmically learning good and interpretable CVs from model simulations is presented. This method permits to assess the quality of the learned CVs and to infer their relation to topological features of the network. In combination with established techniques for learning dynamics from data, an automatic evaluation of the collective behavior can be achieved. This is demonstrated for the CNVM on scale-free networks.

Contents

1. Introduction

Systems containing a large number of interacting entities are the subject of *agent-based modeling*. The relationships between these entities are commonly defined with the help of a network, in which case the model is referred to as a *dynamical system on a network*. Due to their numerous applications, these systems attract the attention of researchers from various disciplines. However, understanding the *collective behavior* of complex agent-based models is naturally difficult as the multitude of local interactions between the many agents often has consequences on a macroscopic scale that are hard to predict. The main focus of this thesis is to investigate the collective behavior of dynamical systems on networks. This chapter motivates and describes the research questions addressed in this thesis, gives an outline of the thesis, and discusses the main contributions.

Agent-based models. In agent-based models (ABMs) [4, 5, 6, 7] a collection of autonomous decision-making entities called *agents* interact with each other and with a shared environment based on a set of rules. The behavior of each agent may depend on its individual properties, its relations to other agents, and its access to (local) information about the environment. ABMs are also called *microscopic models* as they describe a system in terms of its smallest parts. *Macroscopic models* on the other hand aim to describe a system on a global scale by examining the evolution of macroscopic or aggregated variables like averages or densities, typically in the form of differential equations.

For many real-world phenomena, ABMs provide the most natural modeling framework. They are typically the preferred choice for systems involving a possibly heterogeneous population of behavioral entities that interact in complex ways, e.g., humans, human-controlled organizations, or animals. It is often intuitive to specify the desired rules of the ABM, and realistic behavior of agents can usually be guessed or inferred from data or surveys. Stochasticity can easily be included in ABMs to take account of uncertainty or variability of an agent's behavior. Moreover, an ABM offers a lot of flexibility and versatility as its rules can be modified at will and the complexity of the agents' behaviors and interactions can be varied immensely, from simple particle-like agents whose behavior is predetermined by the physics of the simulation to complex agents that make decisions, learn from their history, and adapt their strategies.

Although the actions of each agent may be intuitive and easy to understand, the macroscopic *emergent* or *collective* behavior of the system that results from the agents' interactions is often complex and unpredictable. Even a small change in the environment or in the behavior of the agents may produce a qualitatively different and unexpected outcome on a global scale. As a consequence, it is inherently difficult to directly define macroscopic models for these types of systems as prior knowledge of the complex emergent behavior would be necessary.

Due to their rule-based and modular structure, ABMs are well suited for numerical simulation and are employed in numerous (scientific) fields and applications. For example, in transportation engineering ABMs are used to model traffic and pedestrian flow. In finance and economy, ABMs are frequently employed to model (stock) markets, supply chains, customer management, and marketing strategies. Engineers model smart power grids or the *Internet of things* within an ABM framework. Another common field of application is computational social science, where opinion dynamics and innovation dynamics are modeled using ABMs. Conceptually similar models are also employed in epidemiology to study the spreading of diseases.

Despite its success, agent-based modeling also comes with several issues. Especially for social systems involving human agents, which exhibit a complex decision-making process and sometimes act irrationally, the model parameters and the agents' exact behavior are hard to quantify and verify. Although the rough mechanisms of the underlying system are easily included in an ABM, specific parameter values can often only be guessed. Hence, in most cases an ABM is restricted to providing a qualitative insight into real-world processes; it is not able to produce a precise quantitative prediction. Furthermore, the simulation time and memory requirements of ABMs typically scale at least linearly with the number of agents. Thus, a large scale ABM containing thousands or millions of agents can quickly become computationally intractable. To handle models of that amplitude, one can try to find a reduced representation of the dynamics that keeps the interesting emergent behavior intact, i.e., a fitting macroscopic model. Finding this reduced representation for certain types of models is the main focus of this thesis.

Motivation and research questions. In this thesis, ABMs in the form of *dynamical systems on networks* are considered. Each node of the network represents an agent and the edges between nodes represent some kind of relation between the adjacent agents. Additionally, each agent has a *state* that evolves over time depending on the states of its neighbors in the network. Systems of this type are used across many disciplines, but they are especially prominent in computational social sciences and epidemiology. The objective of this work is to explore the macroscopic behavior of dynamical systems on networks in two steps.

The first step addresses the question of finding a projection of the high-dimensional microscopic system state containing the state of each agent into a low-dimensional space that describes the macroscopic state of the system. Ideally, these projections filter out unnecessary degrees of freedom and quickly decaying processes of the original system while still allowing a description of the essential dynamics, in which case they are called *collective variables* (CVs) or *reaction coordinates*. The mere knowledge of good CVs can already provide a decent understanding of the macroscopic behavior as they describe the information most relevant to the system's dynamics. The aim of this thesis is to find methods for discovering CVs for dynamical systems on networks and to investigate under what conditions certain types of CVs work well.

The second step is concerned with actually finding a reduced macroscopic system, i.e., evolution equations for the macroscopic state given by the collective variables such that the reduced model approximates the low-dimensional projection of the original dynamics. Although the reduced system is generally still stochastic, in some cases the small random actions of many agents may cancel each other out such that, in the fashion of the law of large numbers, the macroscopic dynamics is approximately deterministic. This is referred to as a *concentration effect*. A substantial part of the thesis studies conditions on the system and on the network under which these concentration effects are observed, and the derivation of the resulting deterministic macroscopic dynamics in the form of an ordinary differential equation (ODE). Moreover, macroscopic dynamics in the form of stochastic differential equations (SDEs) and partial differential equations (PDEs) are also considered. The benefits of obtaining a macroscopic system in such a simple form are obvious: a large body of theory and tools designed for studying differential equations can be applied to examine the macroscopic dynamics, e.g., analytical solution methods, stability analysis, and bifurcation theory.

Consider for example the so-called SIS model from epidemiology. In the SIS model on a network each node is either susceptible (S) or infectious (I). Assume that a susceptible node randomly switches to being infectious at the rate $\lambda > 0$ times the percentage of infectious nodes in its neighborhood. For instance, if a susceptible node has two susceptible neighbors and one infectious neighbor, it becomes infectious at the rate $\frac{1}{3}\lambda$. Moreover, assume that infectious nodes randomly become susceptible again at the rate 1. With the theory discussed in this thesis it is possible to show that, for specific networks that satisfy certain conditions, the share of infectious nodes $c(t) \in [0, 1]$ is a collective variable and evolves according to the simple ODE

$$\frac{\mathrm{d}}{\mathrm{d}t} c(t) = -c(t) + \lambda(1 - c(t))\, c(t) \tag{1.1}$$

in the limit of infinitely many agents. For a finite but large number of agents, the above equation is a good approximation of the macroscopic dynamics. Note the *transcritical bifurcation* at the critical value $\lambda_c = 1$: for $\lambda < \lambda_c$ the equilibrium $c = 0$ is stable, but for $\lambda > \lambda_c$ it is unstable and the equilibrium $c_\infty = (\lambda - 1)/\lambda$ is stable instead. As a result, the disease will always die out if the infection rate is smaller than λ_c. If it is larger than λ_c, the share of infectious agents converges to the steady-state c_∞, i.e., the disease becomes endemic in the population. Hence, the critical value λ_c is also called the *epidemic threshold*. This example illustrates how the derivation of a simple reduced dynamical model allows understanding the macroscopic behavior of a complex ABM by applying classical tools, in this case bifurcation theory for ODEs.

Even if the reduced macroscopic system can not be treated easily with classical theory because it is still too complicated, it is nevertheless much faster to evaluate or simulate than the original microscopic model. The reason is that the reduced macroscopic system typically has a much more favorable scaling than the microscopic model, for which the simulation cost increases at least linearly with the number of agents. Hence, knowledge of an approximating reduced system is crucial for understanding the macroscopic behavior without the need for countless hours of simulation of the agent-based model. It is important especially if many simulations of the model have to be conducted for a wide range of parameters to obtain significant statistics, for example when evaluating the effect of different countermeasures in an epidemiological model.

Thesis outline. The following two chapters, chapter 2 and 3, provide an introduction to the topic and mainly contain general considerations, examples, and an overview of related literature. The specific results of this thesis are then discussed in the subsequent chapters 4 to 7. The contents of each chapter are summarized below.

An introduction to dynamical systems on networks is offered in chapter 2. Commonly studied models and their behavior are presented, including the *continuous-time noisy voter model* (CNVM) that acts as guiding example throughout the thesis. Moreover, the concepts of *collective variables* and *concentration effects* are discussed in the context of dynamical systems on networks. Finally, popular model reduction techniques for dynamical systems on networks are introduced.

Random graphs, which are discussed in chapter 3, are an important tool for generating networks with certain characteristics. Common random graph models and their properties are presented. Moreover, a concept called *graphons*, which is useful for studying large dense networks and defining their graph limit, is introduced.

Many popular dynamical systems on networks fall into the class of Markov jump processes, where nodes switch between discrete states according to a continuous-time Markov chain. As it is crucial to be able to examine their behavior and verify theoretical considerations numerically, even for a large number of agents, chapter 4 discusses how to simulate such systems efficiently on a computer. A simulation algorithm for the CNVM, which is also a Markov jump process, is derived and its performance investigated.

For such Markov jump processes the shares of each discrete state in the system, e.g., the percentage of infectious agents in an epidemiological model, are obvious candidates for collective variables. Chapter 5 addresses the question whether certain Markov jump processes on networks possess a mean-field limit in the sense that the stochastic dynamics of the shares of each discrete state concentrate around an ODE in the large population limit. A main theorem that provides conditions for the convergence to this mean-field limit is proven and the rate of convergence is studied. Moreover, this main theorem is applied to the CNVM on several random graph models. For each of these graph models, bounds for the specific parameters that guarantee the convergence to the mean-field limit are derived. The mean-field limit only provides a reasonable approximation to the macroscopic dynamics if the number of nodes is quite large. However, it can be augmented with a stochastic term such that the resulting SDE is able to approximate the dynamics well even for medium-sized populations, which is also demonstrated in this chapter. Finally, an extension of the large population limit to leader-follower models, in which a few very influential agents affect a large mass of ordinary agents, is presented. The resulting reduced model has the form of a piecewise-deterministic Markov process (PDMP).

In chapter 5 it becomes apparent that these mean-field limits of the state shares are valid only in the case of large and homogeneous networks of sufficient density. For networks where the nodes have substantially varying degrees, e.g., scale-free networks, or for sparse networks, the simple state shares are typically not a good choice of collective variables as they do not carry sufficient information to describe the macroscopic dynamics. Thus, in chapter 6 a data-driven method for automatically learning good collective variables for discrete-state dynamical systems on networks is presented. It is applied to several examples and a technique for the numerical validation of the results is discussed. While this method is able to provide good collective variables, it does not generate the reduced system. However, existing data-driven techniques for learning dynamics can be appended to form a complete model reduction pipeline, which is demonstrated for the CNVM on scale-free networks.

Chapter 7 contains several considerations regarding ring-shaped networks, i.e., the one-dimensional periodic lattice and similar networks. The study of dynamical systems on such sparse networks is much harder and generally requires different approaches than the mean-field theory discussed in previous chapters. Instead of dealing with sparse large population limits, this chapter focuses on approximations for the finite ring and the connections to mean-field theory for dense networks. Firstly, it is demonstrated that the mean-field limit for dense networks is unsurprisingly a poor approximation for dynamics on the ring. Moreover, the approximation error of the mean-field limit is examined when interpolating between a ring network and a homogeneous random graph, i.e., in the so-called *small-world regime*. Secondly, a so-called *triplet approximation* is derived for the finite ring, and a novel data-driven approach to reduce the error of the associated *moment closure* is presented. Thirdly, it is demonstrated that, when considering a dense version of the ring graph, graphon theory can be applied to obtain a mean-field limit in the form of a PDE describing the evolution of probability densities in the system. However, it is shown that this graphon mean-field limit fails to produce a good approximation if the graph becomes too sparse.

Finally, the thesis is concluded in chapter 8 where the most relevant aspects are summarized and possible further research questions are discussed.

Main contributions. The two main contributions of this thesis are the considerations about mean-field limits of Markov jump processes on networks presented in chapter 5 and the data-driven algorithm for learning collective variables presented in chapter 6.

The main theorem for mean-field limits in section 5.1, whose proof was developed mainly by the author, provides the foundation to derive bounds for the parameters of several random graph models that guarantee a convergence to the mean-field limit, see section 5.2. Note that sections 5.1 and 5.2 are motivated by and partially reprinted from the author's publication

> [1]: M. Lücke, J. Heitzig, P. Koltai, N. Molkenthin, and S. Winkelmann. *Large population limits of markov processes on random networks.* Stochastic Processes and their Applications, 166:104220, 2023.

with the permission of the co-authors and the publisher. The remaining sections in chapter 5 also build on this main theorem, for example to construct a stochastic limit equation for medium-sized populations.

The data-driven algorithm presented in chapter 6 enables an automatic learning of good collective variables for discrete-state dynamical systems on networks, which is especially useful for network topologies where the mean-field results mentioned previously do not apply. Sections 6.1 and 6.2 are motivated by and partially reprinted from the author's publication

> [2]: M. Lücke, S. Winkelmann, J. Heitzig, N. Molkenthin, and P. Koltai. *Learning interpretable collective variables for spreading processes on networks.* Physical Review E, 109(2):L022301, 2024.

with permission of the co-authors and the publisher. In section 6.3 it is demonstrated that this method can be used to construct a complete data-driven model reduction pipeline by combining it with existing algorithms for learning dynamics. The thereby learned reduced model for the CNVM on scale-free networks is a modification of the standard mean-field limit, which, to the knowledge of the author, has not been discussed in the literature before.

Another contribution of this thesis is the idea to complement moment closure methods with data-driven techniques to minimize the error, which is demonstrated for a triplet approximation of the continuous-time noisy voter model in section 7.2. The approach is largely adopted from the author's publication

> [3]: M. Lücke, P. Koltai, S. Winkelmann, N. Molkenthin, and J. Heitzig. *Discovering collective variable dynamics of agent-based models.* In Extended Abstracts presented at the 25th International Symposium on Mathematical Theory of Networks and Systems MTNS 2022, pages 202-205. University of Bayreuth, 2022.

with permission of the co-authors.

Finally, note that the simulation algorithm for the continuous-time noisy voter model discussed in section 4.2 is implemented in the Python package

> *Spreading Processes on Networks (SPoNet)*, available at
> `https://github.com/lueckem/SPoNet`,

which was developed by the author and is used for all simulations presented in this thesis.

1.1. Remarks on notation

Consider the following remarks:

- The terms *network* and *graph* are used synonymously to describe a pair $G = (V, E)$. The elements of V are commonly called *vertices*, *nodes*, *sites*, or *points*. In this thesis the term *nodes* is used and in the context of dynamical systems on networks the nodes are also referred to as *agents*. The set E contains pairs of nodes that are commonly called *edges*, *links*, *bonds*, or *lines*. In this thesis the term *edges* is used.

- Random variables are set in bold typeface, e.g., $\boldsymbol{x}$ may denote a random variable and x a realization of $\boldsymbol{x}$. The notation $\boldsymbol{x} \in \mathbb{X}$ is employed to specify that $\boldsymbol{x}$ is a $\mathbb{X}$-valued random variable.

- For two functions $f, g : \mathbb{R} \to \mathbb{R}_{>0}$ the asymptotic dominance of f over g is denoted by
$$f(x) \gg g(x) \quad :\Leftrightarrow \quad f(x) = \omega(g(x)) \quad :\Leftrightarrow \quad \lim_{x\to\infty} \frac{f(x)}{g(x)} = \infty \tag{1.2}$$
and asymptotic dominance of g over f by
$$f(x) \ll g(x) \quad :\Leftrightarrow \quad f(x) = o(g(x)) \quad :\Leftrightarrow \quad \lim_{x\to\infty} \frac{f(x)}{g(x)} = 0. \tag{1.3}$$

- Given a number $N \in \mathbb{N}$, the abbreviation $[N] := \{1, \dots, N\}$ is employed.

2. Dynamical systems on networks

In this chapter an introduction to the types of dynamical systems considered in this thesis will be presented, as well as an overview of common example systems and their behavior. Furthermore, the concepts of *collective variables* and *concentration effects* will be introduced in the context of dynamical systems on networks. Finally, some popular model reduction techniques for such dynamical systems are discussed and their relation to the notion of collective variables explained.

Let a simple graph $G = (V, E)$ with $N \in \mathbb{N}$ nodes be given, where $V := [N]$ is the set of nodes and E the set of edges. The symmetric *adjacency matrix* $A \in \{0, 1\}^{N \times N}$ encodes which edges are present and thus provides an equivalent representation of the graph G. Every node $i \in V$ is equipped with a state $x_i \in \mathbb{X}$, where $\mathbb{X}$ is the set of all possible states, and the collection $x := (x_1, \dots, x_N) \in \mathbb{X}^N$ is called the *system state*. A dynamical system on this network G describes how the state x_i of each node changes over time, due to an inherent dynamics and due to the influence of neighboring nodes [8, 9]. The system may follow a deterministic law given by differential equations or update equations, or it may be defined in the form of a stochastic process, in which case it is also commonly called an *interacting particle system* [10].

In this setup, each node can be viewed as an *agent* and the edges as some form of (social) interaction or relation between agents. The dynamics represents the behavior of each agent, given its local interactions via the edges to neighboring agents. Thus, such a dynamical system on a network can be considered as a special type of *agent-based model*.

This thesis mainly considers *simple graphs*, where each edge is undirected and connects exactly two nodes, and self-loops and multiple edges between two nodes are not allowed. Moreover, the network is assumed to be static, i.e., it does not change over time. These restrictions make a mathematical analysis more feasible and there are still numerous interesting and complex systems that fall into this category, see section 2.1 for some examples. Extending this framework often leads to various mathematical and practical issues, making the system substantially more difficult to analyze. The most commonly discussed extensions are presented below. Due to their additional difficulty, these extensions will not be considered in this thesis.

In *higher-order networks* or *hypergraphs*, an edge may join any number of nodes, and hence they allow modeling complex interactions that require a group of three or more agents, see [11] for an overview.

If the network structure evolves in dependence on the nodes' states and in turn the nodes' states are influenced by the network structure, the system is often called an *(adaptive) coevolutionary network* [12, 13, 14]. This interplay between network dynamics and state dynamics results in many interesting phenomena like self-organization or the formation of polarized clusters, but comes at the price of tremendous additional mathematical (and numerical) challenges.

Finally, allowing directed or weighted edges adds the possibility of one-sided interactions or interactions of different strengths. However, a similar effect can be achieved by considering an undirected graph and including the strength at which agents influence each other in the dynamical model. Many example systems and techniques that are discussed later for undirected graphs can also easily be extended to directed or weighted graphs.

This chapter is structured as follows. Some of the most commonly studied examples of dynamical systems on networks are presented in section 2.1. Section 2.2 contains a detailed introduction to the *continuous-time noisy voter model*, which will serve as a guiding example throughout this thesis. The concepts of *collective variables* and *concentration effects* are discussed in section 2.3. Finally, common techniques for the model reduction of dynamical systems on networks are presented in section 2.4, i.e., mean-field limits, moment closure methods, and (automorphism-based) lumping.

2.1. An overview

In this section an overview of some of the most studied dynamical systems on networks is presented. This overview is by no means an exhaustive survey, but gives an introduction to a few selected prominent models or types of models. While there is an abundance of different dynamical systems on networks that are studied in the context of numerous applications, many of them are extensions of a small selection of basic models that cover the most fundamental mechanisms. Extensions of these basic models are often developed by introducing stochasticity (noise), external perturbations, a heterogeneous population, a higher-dimensional system state, or more complex behavioral rules for the agents. Even small modifications to a model can result in a qualitatively different collective behavior and thus the investigation of this bulk of different models has been (and still is) of large interest in many scientific communities. For instance, in discrete-time models a different update order of nodes, e.g., *synchronous* updating (all nodes change their state in one step) instead of *asynchronous* updating (one random node changes its state in one step), may already produce a different outcome [8]. Still, the basic models described below provide an overview of common mechanisms and their resulting behaviors that build the foundation of understanding more complex models.

We first present three continuous-state models in section 2.1.1: systems of coupled ODEs, the DeGroot model, and bounded confidence models. Then three types of discrete-state models are discussed in section 2.1.2: voter models, threshold models, and random walks.

2.1.1. Continuous-state models

Systems of coupled ODEs. Many popular continuous-state dynamical systems on networks have the form of a system of coupled ODEs, i.e., the state $x_i \in \mathbb{R}$ of node i evolves according to the ODE

$$\frac{\mathrm{d}}{\mathrm{d}t} x_i(t) = f_i(x_i(t)) + \sum_{j=1}^{N} A_{i,j} \; g_{i,j}(x_i(t), x_j(t)), \tag{2.1}$$

where the function f_i describes the inherent dynamics of node i, A is the adjacency matrix of the underlying network, and the function $g_{i,j}$ describes the influence of node j on node i. Some models assume a simplified setup, in which every node is subject to the same internal dynamics $f_i = f$, or interactions are symmetrical and identical for every pair, i.e., $g_{i,j} = g$

for all i, j. Due to the generality of this model description, it is employed in numerous biological, technological, and social systems, see [15] for an overview. It can be used to study a variety of complex behaviors like spreading and propagation, cascades, critical transitions, collective behavior, organization, and synchronization [16].

An important example of a system of the above form is the *Kuramoto model* [17], which describes the synchronization of oscillators. The state of the i-th oscillator is given by its phase θ_i, and its natural frequency is denoted by ω_i. The oscillators are coupled such that they influence each other based on the sine of their phase differences, i.e., each oscillator follows the ODE

$$\frac{\mathrm{d}}{\mathrm{d}t}\theta_i(t) = \omega_i + \sum_{j=1}^{N} A_{i,j}\,\lambda_{i,j}\,\sin(\theta_j(t) - \theta_i(t)), \tag{2.2}$$

where $\lambda_{i,j}$ denotes the coupling strength between oscillators i and j. Depending on the network topology and the coupling strengths, the system may converge to a fully synchronized state such that all oscillators rotate at the same frequency eventually (despite having different natural frequencies). For smaller coupling strengths a partly or completely disordered (incoherent) system state may be observed. Studying the phase transitions between these regimes is still a focus of ongoing research [18].

The DeGroot model. In the *(social network) DeGroot model* [19] each agent i has an opinion $x_i \in [0,1]$ about a topic. It is convenient to describe the relations between agents as a *trust matrix* $T \in [0,1]^{N\times N}$, where $T_{i,j}$ represents the weight that agent i puts on agent j's opinion. (T can equivalently be thought of as the adjacency matrix of an underlying weighted directed graph.) The trust matrix is stochastic, i.e., its rows sum to 1. Then the state x evolves according to the discrete-time update rule

$$x(t) = Tx(t-1) = T^t x(0). \tag{2.3}$$

From the perspective of opinion dynamics, an important issue to investigate is the occurrence of *consensus*, that is, whether all agents converge to the same opinion. For example, it has been shown from standard results in Markov chain theory that if the associated graph is strongly connected and aperiodic, a consensus is always reached [20]. If on the other hand the graph is not aperiodic, a periodic dynamics may be observed.

The DeGroot model can also be viewed as a model of a learning process within a population. For instance, the initial state $x_i(0)$ of each agent could be given by a noisy signal about some true value. Under the right conditions [20], the agents then collectively converge to the true value during the update process ("wisdom of crowds").

Bounded confidence models. The central characteristic of *bounded confidence models* in opinion dynamics is that agents only interact if they already have a similar opinion. Agents that fundamentally disagree with each other are not willing to compromise and hence do not interact. Bounded confidence models are motivated by a sociological phenomenon called *homophily*, which describes the tendency of individuals to prefer associating and interacting with similar others [21]. The two most popular bounded confidence models are the *Deffuant model* and the *Hegselmann–Krause model.*

In the Deffuant model [22] each agent i has an opinion $x_i \in [0,1]$ about a topic. Two agents i and j can interact with each other if they are neighbors in the underlying graph and if the difference between their opinions does not exceed a threshold $\varepsilon \in [0,1]$. The dynamics of the Deffuant model follows a stochastic discrete-time update step, in which first a random pair of two neighboring agents i and j is picked. If $|\boldsymbol{x}_i(t) - \boldsymbol{x}_j(t)| > \varepsilon$, nothing happens. If however $|\boldsymbol{x}_i(t) - \boldsymbol{x}_j(t)| \leq \varepsilon$, then the opinions of agents i and j are updated as follows:

$$\boldsymbol{x}_i(t+1) = \boldsymbol{x}_i(t) + \mu[\boldsymbol{x}_j(t) - \boldsymbol{x}_i(t)] \tag{2.4}$$

$$\boldsymbol{x}_j(t+1) = \boldsymbol{x}_j(t) + \mu[\boldsymbol{x}_i(t) - \boldsymbol{x}_j(t)] \tag{2.5}$$

where $\mu \in [0, \frac{1}{2}]$ is called the *convergence parameter*. Thus, agents i and j will compromise so that after the update their opinions are closer together than before. In the extreme case $\mu = \frac{1}{2}$ both agents adopt the average of the two opinions. The opinions of all other agents $k \notin \{i, j\}$ do not change in this update step, i.e., $\boldsymbol{x}_k(t+1) = \boldsymbol{x}_k(t)$.

The characteristic behavior of the Deffuant model involves the formation of polarized clusters of agents. All agents inside a cluster converge to the same opinion and the opinion differences between clusters are sufficiently large so that there is no interaction between them [23]. Even though the update is stochastic, this state of polarized clusters constitutes a stable equilibrium of the system. Given a connected network and the choice $\varepsilon = 1$, the model always converges to a state of consensus where every node has the same opinion. For smaller values of ε, a larger number of polarized clusters will form. A more precise quantification of the number of clusters and the rate at which they form, depending on the underlying graph and the initial conditions, can be found in [23]. However, the Deffuant model and its variations are still not completely understood and continue to be the subject of recent research.

The *Hegselmann–Krause* model [24] is similar to the Deffuant model but employs a slightly different update rule in which an agent adapts to all opinions in its neighborhood instead of compromising only with one adjacent agent. In the update step, a random agent i is picked and its opinion set to the average of all suitable agents in its neighborhood

$$\boldsymbol{x}_i(t+1) = \frac{\sum_{j=1}^{N} \delta_{|\boldsymbol{x}_i(t) - \boldsymbol{x}_j(t)| \leq \varepsilon} \, A_{i,j} \, \boldsymbol{x}_j(t)}{\sum_{j=1}^{N} \delta_{|\boldsymbol{x}_i(t) - \boldsymbol{x}_j(t)| \leq \varepsilon} \, A_{i,j}}. \tag{2.6}$$

Again, only interactions with agents of a sufficiently similar opinion are considered due to the factor $\delta_{|\boldsymbol{x}_i(t) - \boldsymbol{x}_j(t)| \leq \varepsilon} \in \{0, 1\}$. This dynamics generally behaves similar to the Deffuant model and also results in the characteristic formation of polarized opinion clusters, see [23] for an overview.

A continuous-time version of the bounded confidence model can be defined by letting each connected pair that has sufficiently similar opinions compromise continuously, which leads to a system of coupled ODEs [25]

$$\frac{\mathrm{d}}{\mathrm{d}t} x_i(t) = \frac{1}{N} \sum_{j=1}^{N} A_{i,j} \; w(x_j(t) - x_i(t)), \quad w(\xi) := \begin{cases} \xi, & |\xi| \leq \varepsilon \\ 0, & |\xi| > \varepsilon. \end{cases} \tag{2.7}$$

Alternatively, a stochastic continuous-time version can be formulated by drawing the time points of interactions from exponential distributions.

2.1.2. Discrete-state models

Voter models. The so-called *voter model* [26, 27] is one of the most studied models of opinion dynamics. Each agent i has a discrete opinion $x_i \in \{0,1\}$ that could for instance describe which of two political parties the agent intends to vote for. There are both discrete-time and continuous-time versions of the voter model. In the discrete-time version, a random agent i is picked in each step. The agent i then chooses a random neighbor j and adopts its opinion, i.e., $\boldsymbol{x}_i(t+1) = \boldsymbol{x}_j(t)$. The states of all other agents are not changed in this update step. Due to this extremely simple imitation (or infection) mechanism, the voter model is often considered as a paradigmatic model of a spreading process on a network. Hence, it is not limited to opinion dynamics, but has also been applied in numerous other applications related to spreading or diffusion phenomena (on networks) [23]. Moreover, in many cases this simplicity enables an extensive mathematical analysis, which has led to a great deal of research on its dynamics. However, it is also questionable if the voter model is powerful enough to actually describe the behavior of many real-world processes, especially involving complex issues like political voting [28]. Due to these concerns, many extensions and variations of the voter model have been developed to better fit the mechanisms observed in certain applications, see [23, 8] for an overview.

Given the update mechanism of the voter model, it is easy to see that the two *consensus* system states, i.e., $x_i = 0$ for all i or $x_i = 1$ for all i, are absorbing states. Hence, on any finite-size connected network the voter model will at some point reach consensus and stay like that forever. Until consensus however, the dynamics of the voter model is complex and interesting, and heavily depends on the underlying network structure. For example, on so-called small-world networks (see section 3.1.4 for details) the voter model is stuck in a metastable state of coexistence, where the shares of both opinions stay almost constant for a long time, before suddenly one opinion dies out and consensus is reached [23]. The voter model is also often studied on infinite-size networks like lattices, for which it is possible that the two opinions coexist indefinitely [10].

The discrete-time voter model as introduced above is often referred to as the *direct* voter model. A different popular variation is the *reverse* voter model, in which a random agent i and a random neighbor j of agent i are picked as well, but then agent j adopts the opinion of agent i instead of the other way around. Moreover, in the *link-update* voter model, a random edge $\{i,j\}$ is picked and then a coin toss decides whether agent i adopts the opinion of agent j or vice versa. Depending on the underlying network topology, these slight variations of the update process can already result in substantially different (emergent) dynamics [23].

In the continuous-time version of the voter model, each agent starts a Poisson clock which notifies the agent after a random exponentially distributed time has passed. If an agent is notified by its clock, it adopts the opinion of a random neighbor and resets its clock. Hence, the overall system can be described as a continuous-time Markov chain. By employing different ways to calculate the transition rates in that Markov chain based on the neighborhoods of agents, a whole class of diverse voter models can be defined [10]. In this thesis such a variant, which is referred to as the *continuous-time noisy voter model* (CNVM), will serve as a guiding example and will be examined in great detail. The CNVM is introduced in section 2.2.

Majority rule and threshold models. The sociological phenomenon of *conformity* describes how individuals tend to adjust their beliefs to match prevalent social norms in their social group, and is the motivation for *threshold models* [29]. The central mechanism in threshold models is that agents will change their opinion if and only if a sufficient number (or fraction) of their neighbors share this new opinion. Otherwise, agents conform to the status quo and keep their original opinion. For example, in the *Watts model* [30] each node i has an opinion $x_i \in \{0,1\}$ and a fixed threshold $r_i \in [0,1]$. Typically, r_i is drawn randomly from some distribution. Nodes with state $x_i = 0$ are called *inactive*, and nodes with state $x_i = 1$ *active*. In the update step of the Watts model, a random node i is picked and if it is not active, it becomes active if the fraction of its active neighbors exceeds its threshold r_i. If node i was already active, its state remains unchanged. Hence, the Watts model is monotonic in the sense that agents can never become inactive again. It has been used to study the occurrence of *global cascades*, i.e., the collective behavior that a large number of inactive nodes suddenly become active after a small perturbation or shock [30]. A variety of different (non-monotonic) threshold models have been proposed and studied, see [8] for an overview.

In *majority rule models* [31, 32] agents update their state to the majority state in their neighborhood as if they were overruled by their neighbors. Thus, majority rule models can be considered as a special case of threshold model in which every agent has the same threshold $r_i = 0.5$. However, the term *threshold model* is more frequently used in applications involving spreading phenomena, whereas the term *majority rule model* is preferred for models describing decision-making or voting processes.

Random walks and exclusion processes. In all previously presented models the nodes of the graph represented agents that are connected to each other by the network edges and have a state that evolves over time. In the context of *random walks* and *exclusion processes* on networks the interpretation is fundamentally different. Here, the nodes of the graph represent *sites* that an agent can occupy. The state x_i of node i describes its occupation status, for example $x_i = 1$ if it is currently occupied by an agent and $x_i = 0$ if it is free. If two sites i and j are connected by an edge, an agent can travel between them such that its destination site becomes occupied while its origin site is freed. Hence, an agent takes a "walk" on the network by traveling from one node to the other, and the system state x describes which nodes are currently occupied by agents.

The most studied model of this type is the *random walk* [33, 34], in which a single agent travels across the network by jumping to a uniformly random neighboring node in every discrete time step. The random walk on networks has applications in various fields, one of the most noteworthy being the ranking of websites for internet search engines [35]. If several agents traverse the network simultaneously with the restriction that each site can only hold one agent at the same time, the model is referred to as an *exclusion process* [10, 36]. Continuous-state versions of these types of dynamics, where some continuous quantity flows between the nodes, are often referred to as *transport processes* on networks, see [37, 38] for details.

Since this thesis is more focused on dynamical systems in the context of the former interpretation that the nodes represent the agents and the system state x contains the state of each agent, random walks on networks and transport processes will not be discussed further here.

2.2. The continuous-time noisy voter model

In the *continuous-time noisy voter model* (CNVM) each node $i \in [N]$ of the simple graph has one of $M \in \mathbb{N}$ different opinions $x_i \in [M]$. As before, the system state $x = (x_1, \dots, x_N) \in [M]^N$ consists of the opinion of each node. The opinion of node i changes over time in the form of a continuous-time Markov chain, such that the transition rate matrix $Q_i(x) \in \mathbb{R}^{M \times M}$ depends on the opinions of neighbors of node i, which is denoted by the dependence of Q_i on x. (Hence, the Markov chain induced by Q_i is non-homogeneous in the sense that the transition rates do not remain constant.) The rate at which node i switches from one opinion $m \in [M]$ to a different opinion $n \in [M]$ is defined as

$$(Q_i(x))_{m,n} \coloneqq r_{m,n} \frac{d_{i,n}(x)}{(d_i)^\alpha} + \tilde{r}_{m,n}, \tag{2.8}$$

where $d_{i,n}(x)$ denotes the number of neighbors of node i with opinion n, d_i is the degree of node i, and $r_{m,n}, \tilde{r}_{m,n} \geq 0$ and $\alpha \in \mathbb{R}$ are model parameters. Thus, the transition rate consists of two terms. The first term, $r_{m,n}\, d_{i,n}(x)/(d_i)^\alpha$, describes the influence of its neighborhood on node i, or equivalently the rate at which node i imitates (copies) its neighbors. The more neighbors of agent i have opinion n, the higher is the rate at which it transitions to opinion n. The second term, $\tilde{r}_{m,n}$, is an additional transition rate that is not dependent on the neighborhood. It results in "explorative" behavior of agents, which can also be interpreted as *noise*.

The model parameter α controls the relation between the degree of a node and the influencing force it experiences from its neighborhood. In particular the choices $\alpha = 0$ and $\alpha = 1$ allow a simple interpretation of the underlying mechanisms. For $\alpha = 0$, the transition rates to opinion n are proportional to the *number* of neighbors of opinion n. This is for example a popular choice for infection modeling [39] as then each link to a neighboring node potentially provides a cumulative increase in the infection rate. Nodes with large degrees have the potential to be exposed to large transition rates, whereas nodes with small degrees are rather inert.

The choice $\alpha = 1$ on the other hand results in transition rates to opinion n that are proportional to the *share* of neighbors of opinion n. To interpret the underlying interaction mechanism, it is advantageous to rewrite the model parameters as $r_{m,n} = \hat{r}\, p_{m,n}$, where $\hat{r} \geq 0$ and $p_{m,n} \in [0,1]$. (For example, $\hat{r} \coloneqq \max_{m,n} r_{m,n}$ and $p_{m,n} \coloneqq r_{m,n}/\hat{r}$.) Then the model reproduces the following procedure:

1. Each agent starts a Poisson clock that notifies the agent after a random, exponentially distributed with parameter $\hat{r}$, time has passed.

2. If an agent i is notified by its clock, it picks a random neighbor for interaction. Hence, the probability to interact with a neighbor of opinion n is $d_{i,n}(x)/d_i$.

3. After the interaction, agent i adopts the opinion n of the chosen neighbor with probability $p_{m,n}$, and restarts its clock.

Thus, for $\alpha = 1$ all nodes undergo transitions at the same rate, irrespective of their degree. If not stated otherwise in the text, the default parameter $\alpha = 1$ is used in the CNVM.

Although the local continuous-time Markov chains describing the behavior of individual agents are non-homogeneous, the global process, i.e., the evolution of the system state x, can be described as a homogeneous continuous-time Markov chain on the state space $\mathbb{X} = [M]^N$. The generator Q of this global process is given by

$$Q(x, y) = \begin{cases} \big(Q_i(x)\big)_{x_i, y_i} & \exists i \in [N]\ \forall j \neq i : x_j = y_j \\ 0 & \text{else} \end{cases} \tag{2.9}$$

and specifies the rate of transitioning to a state $y \in [M]^N$ when starting in a different state $x \in [M]^N$.

Remark 2.1. In the case of a complete graph, the CNVM is equivalent to a *well-mixed chemically reacting system* [40, 41], in which the nodes now represent particles in a well-stirred solution. Instead of describing an opinion, the state denotes that a particle belongs to one of M chemical *species* that can undergo certain reactions. In the CNVM there are two types of reactions: a particle of species A reacts with a different particle of species B to form two particles of species B (imitation), typically written as $A + B \to 2B$, and a particle of species A spontaneously transitions to species B (noise/exploration), typically written as $A \to B$. The term *well-mixed* implies that each particle has the same probability to interact with any other particle, hence the complete graph. However, for a non-complete graph that restricts which nodes can interact, the behavior of the CNVM can become much more complex than what is observed in well-mixed chemically reacting systems.

Many popular epidemiological models can be represented in the form of the CNVM as the following example suggests.

Example 2.2 (SIS & SIR model)**.** In epidemiology, the *SIS model* describes the spreading of a disease on a network [39]. Each node i is either susceptible ($x_i = 1$) or infectious ($x_i = 2$). For each neighboring infectious node, a susceptible node also becomes infectious at the infection rate $\lambda > 0$. An infectious node recovers at the rate 1 and becomes susceptible again. Thus, the SIS model is a special case of the CNVM with model parameters

$$r = \begin{pmatrix} - & \lambda \\ 0 & - \end{pmatrix}, \quad \tilde{r} = \begin{pmatrix} - & 0 \\ 1 & - \end{pmatrix} \tag{2.10}$$

and (typically) $\alpha = 0$.

A popular modification of this model is that infectious nodes become immune to the disease after recovering, i.e., they are removed from the susceptible population. The resulting *SIR model* is also a special case of the CNVM with an additional third state:

$$r = \begin{pmatrix} - & \lambda & 0 \\ 0 & - & 0 \\ 0 & 0 & - \end{pmatrix}, \quad \tilde{r} = \begin{pmatrix} - & 0 & 0 \\ 0 & - & 1 \\ 0 & 0 & - \end{pmatrix}. \tag{2.11}$$

Characteristic behavior. In the absence of noise ($\tilde{r}_{m,n} = 0$ for all m, n) the consensus states, where all nodes have an identical opinion, are absorbing states. Hence, the CNVM will at some point reach consensus and stay like that indefinitely. If however $\tilde{r}_{m,n} > 0$

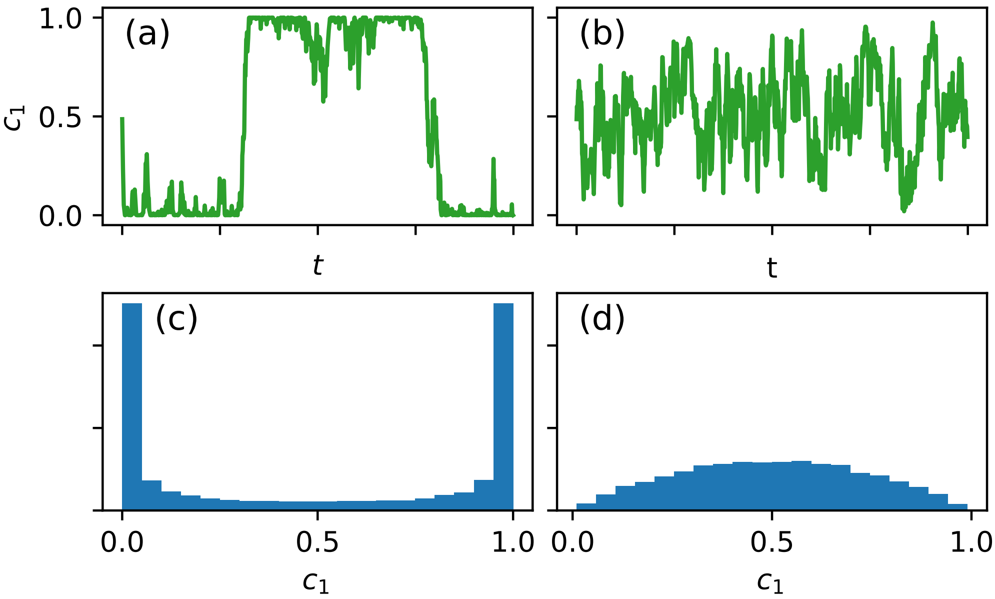


Figure 2.1.: Bimodal to unimodal transition of the invariant distribution due to varying magnitude of noise. Trajectories of the share c_1 of opinion 1 in a low noise (a) and high noise (b) regime, and associated distributions (c) and (d). CNVM with $M = 2$ opinions on Albert–Barabási network (see section 3.1.5 for details) of size $N = 1000$.

for all m, n, the underlying continuous-time Markov chain is irreducible and recurrent (and thus ergodic), which implies the existence of a unique invariant distribution [42]. For many network topologies a characteristic transition from a bimodal to an unimodal invariant distribution can be observed based on the magnitude of the noise parameters $\tilde{r}$ compared to the infection parameters r [43]. In the low noise regime the system spends most time close to the consensus states, with rare transitions between them due to the noise. This metastability results in a bimodal behavior in the sense that the share of each opinion is close to zero or close to one most of the time, see figure 2.1. As the magnitude of noise increases, the mutual influencing between neighbors becomes less important for the system's behavior. Thus, in the high noise regime the states of agents are essentially uncorrelated because their state transitions are dominated by noise. Consequently, the invariant distribution of the opinion shares becomes unimodal, as illustrated in figure 2.1. In this regime, the system is predominantly found in a polarized state where no opinion has the vast majority.

It should be noted that the transition from bimodal to unimodal behavior is a finite-size effect that cannot be observed in the large population limit. For example, it has been shown [43] that on a complete graph the noise threshold between bimodal and unimodal decreases proportionally to $1/N$ as $N \to \infty$ (given fixed infection rates $r_{m,n}$). Hence, even for small amounts of noise, the unimodal regime will be observed in the case of large populations N. Furthermore, the variance of the invariant distribution also decreases proportionally to $1/N$ as $N \to \infty$. As a result, for large N the trajectories of the opinion shares converge to a (almost) deterministic steady state, which implies that in equilibrium the CNVM is found only in states x with essentially identical shares of opinions.

Similar phenomena can be observed for many dynamical models and network topologies. The next section introduces *collective variables* and the *concentration effect* in order to formalize the above observations.

2.3. Collective variables and the concentration effect

Collective Variables. Across many scientific fields it is observed that high-dimensional and (seemingly) extremely complex dynamical systems actually exhibit an underlying regularity such that their behavior is much simpler than the large number of degrees of freedom would allow for. *Collective variables* (CVs) are projections from the original high-dimensional system state into a low-dimensional reduced space that extract exactly this regularity while filtering out the unnecessary degrees of freedom. They allow the definition of a comparatively simple reduced model in this reduced state space that replicates the essential dynamical properties of the original model.

The concept of CVs is widely established in computational physics and chemistry, especially in the field of molecular or atomistic dynamics [44, 45, 46]. For instance, describing the position of each amino acid residue in a large protein results in a high-dimensional and complex system. But, due to physical constraints, many proteins are at most times found in only a few energetically advantageous configurations. Their dynamics is characterized by metastability [47], i.e., they remain in one of the configurations for a long time until suddenly transitioning to a different configuration. In this setting, it is an important task to find CVs that resolve exactly these metastable configurations and the *transition paths* [48] that the protein takes to switch between them. Due to this interpretation of describing the paths between important regions of the state space (e.g., reactant state and product state), CVs are also often called *reaction coordinates* [49, 50].

The theory of CVs, which was largely developed with applications like molecular dynamics in mind, is increasingly being applied in other disciplines as well, for example fluid dynamics [51], coupled oscillators [52, 53], and agent-based models [54, 55]. It is a topic of ongoing research to define CVs in a mathematically abstract manner and to determine what properties of a general dynamical system imply their existence. The approach presented in [56, 57, 58] is outlined in the following.

Let a Markovian stochastic process $\boldsymbol{x}(t)$ on the state space $\mathbb{X} \subset \mathbb{R}^N$ be given. Assume that for each $x \in \mathbb{X}$ and $t > 0$ the *transition density function* $p_x^t : \mathbb{X} \to \mathbb{R}_{\geq 0}$ can be defined as the density (w.r.t. Lebesgue) of the distribution of $\boldsymbol{x}(t)$ given $\boldsymbol{x}(0) = x$. Furthermore, assume that the stochastic process has a unique stationary distribution with density ρ and that it is ergodic in the sense that for every $x \in \mathbb{X}$ the transition density p_x^t converges to ρ as $t \to \infty$. In this setting, a d-dimensional CV is simply a map $\varphi : \mathbb{X} \to \mathbb{R}^d$. However, to fulfil the purpose of a CV, the dimension d should be as small as possible while the map φ still extracts the essential information of the state. More precisely, there should exist functions $\tilde{p}_{\varphi(x)}^t : \mathbb{X} \to \mathbb{R}_{\geq 0}$ such that

$$p_x^t \approx \tilde{p}_{\varphi(x)}^t \tag{2.12}$$

for all $x \in \mathbb{X}$ and times t of intermediate scale. Hence, the distribution of the system only depends on the CV value $\varphi(x)$ of the initial state x, and not the initial state itself. For very small times t with respect to the system's typical timescales, the transition densities p_x^t

are close to Dirac delta distributions at x, and therefore it is never possible to find a low-dimensional CV φ such that (2.12) holds. For very large times on the other hand, any choice of φ is valid in the sense of (2.12) because all p_x^t are close to the stationary density ρ. (For instance, choose $\varphi \equiv 0$ and $\tilde{p}_0^t = \rho$.) Therefore, a *good* CV φ implies that (2.12) holds for the intermediate time regime, such that the fast processes of the system have already equilibrated but it has not yet converged to the stationary distribution.

A desirable property of a good CV is that important dynamical features are preserved when the process is projected down into the CV space. This statement can be specified with the help of the *transfer operator* (also called *Perron–Frobenius operator*) $\mathcal{P}^t : L^1(\mathbb{X}) \to L^1(\mathbb{X})$, which describes the density propagation due to the process $\boldsymbol{x}(t)$, i.e., given an initial distribution over the state space with density u (w.r.t. Lebesgue) at time 0, the system state is distributed according to the density $\mathcal{P}^t u$ at time t. The transfer operator can be defined via the transition density functions

$$(\mathcal{P}^t u)(y) := \int_{\mathbb{X}} u(x)\, p_x^t(y)\, \mathrm{d}x. \tag{2.13}$$

Moreover, the *projection operator* $\Pi_\varphi : L^1(\mathbb{X}) \to L^1(\mathbb{X})$ averages a density u over level sets of the CV φ, such that $\Pi_\varphi u$ is constant on each level set. It can be defined via the conditional expectation

$$(\Pi_\varphi u)(x) = \mathbb{E}_{\boldsymbol{x}\sim\rho}\big[u(\boldsymbol{x}) \mid \varphi(\boldsymbol{x}) = \varphi(x)\big] \tag{2.14}$$

and is similar to the Zwanzig projection operator from statistical physics [59]. Note however that $\Pi_\varphi u$ is still a density over the original state space $\mathbb{X}$, and not over the reduced space $\varphi(\mathbb{X})$. (Nevertheless, the image of Π_φ is isomorphic to $L^1(\varphi(\mathbb{X}))$, which can thus be embedded into $L^1(\mathbb{X})$.) This allows the definition of the *effective transfer operator* $\mathcal{P}_\varphi^t : L^1(\mathbb{X}) \to L^1(\mathbb{X})$, which applies the projection operator to a density, evolves it according to the stochastic process, and then applies the projection operator again:

$$\mathcal{P}_\varphi^t u := \Pi_\varphi\, \mathcal{P}^t\, \Pi_\varphi\, u. \tag{2.15}$$

The effective transfer operator describes the effective dynamics, i.e., the stochastic process projected down into the CV space, but embedded into the original space. This has the advantage that one can directly compare the original dynamics $\mathcal{P}^t$ and the reduced dynamics $\mathcal{P}_\varphi^t$ as they are defined on the same space. In fact, for a *good* CV the original and the effective dynamics match closely, i.e.,

$$\mathcal{P}^t u \approx \mathcal{P}_\varphi^t u \quad \text{for all } u, \tag{2.16}$$

which implies that a reduced model can be defined on the reduced space that is approximately Markovian and replicates the dynamics of the original system.

Finding collective variables. For most applications it is a rather difficult task to determine if low-dimensional good CVs (in the sense of (2.16)) exist, and an even more difficult task to find them. As a consequence, oftentimes reasonable CVs of complex systems are simply guessed by experts due to their knowledge of the underlying mechanisms. For example, in the case of molecular dynamics typical guesses are certain bond angles or residue distances based on the specific protein at hand. While in the past a mathematical justification of these guesses was often missing, a lot of theory and numerical (data-driven) algorithms to find CVs have been developed in recent years.

For instance, in the special case of reversible metastable stochastic processes, it is known that good CVs are given by the dominant eigenfunctions of the transfer operator $\mathcal{P}^t$. Assuming that the spectrum of $\mathcal{P}^t$ is discrete, let λ_i^t denote the eigenvalues of $\mathcal{P}^t$, sorted by decreasing absolute value, and θ_i the associated eigenfunctions. Due to the reversibility of the process, the eigenvalues λ_i^t are real and lie between 0 and 1 [60]. The largest eigenvalue is $\lambda_0^t = 1$ for all t, and the associated eigenfunction is the stationary density ρ. The subsequent eigenvalues go to 0 for both increasing index and increasing time, i.e.,

$$\lim_{i\to\infty} \lambda_i^t = 0 \quad \text{and} \quad \lim_{t\to\infty} \lambda_i^t = 0 \text{ for } i \geq 1. \tag{2.17}$$

Moreover, the metastability of the process implies a spectral gap [61, 62], i.e., there exists a number K such that the K-th largest eigenvalue is much larger than the $(K+1)$-th largest eigenvalue

$$1 = \lambda_0^t > \lambda_1^t \geq \cdots \geq \lambda_K^t \gg \lambda_{K+1}^t \geq \cdots \geq 0. \tag{2.18}$$

The eigenfunctions θ_i however do not depend on t. As a result, the propagation of a density $u = \sum_{i=1}^{\infty} \alpha_i \theta_i$ is given by

$$\mathcal{P}^t u = \sum_{i=0}^{\infty} \lambda_i^t \alpha_i \theta_i \approx \sum_{i=0}^{K} \lambda_i^t \alpha_i \theta_i, \tag{2.19}$$

which has the following interpretation: the stochastic process can be seen as a superposition of decaying processes in the sense that any contribution to the density u that is not aligned with the stationary density $\rho = \theta_0$, but with some other eigenfunction, decays over time, with the rate of decay being larger the smaller the corresponding eigenvalue is. Due to the spectral gap, the density propagation of the system is thus mostly determined by the dominant spectrum of $\mathcal{P}^t$, which is associated to the slowly decaying processes. It is therefore not surprising that the CV containing the dominant eigenfunctions

$$\varphi(x) = \begin{pmatrix} \theta_1(x) \\ \vdots \\ \theta_K(x) \end{pmatrix} \in \mathbb{R}^K \tag{2.20}$$

is good in the sense of (2.16), and the error introduced by the CV is smaller the more pronounced the metastability of the system is, see [57, 58] for details. However, this choice of CV might still not be optimal. For instance, if the system is characterized by only a few transition pathways between the metastable sets, the K dominant eigenfunctions can be parametrized by an even lower-dimensional CV [58, section 6.2].

Due to the significance of the transfer operator spectrum, a large number of (data-driven) methods that focus on its approximation have been presented. They often share many similarities but also exhibit subtle differences, which will not be specified in detail in the following non-exhaustive selection. One of the earliest of these approaches is called *Perron cluster cluster analysis* (PCCA) [63], which allows the clustering or coarse-graining of discrete-state Markov chains based on the spectrum of the transfer operator. Later, algorithms that are also suitable for continuous-state Markov processes have been developed, for instance *extended dynamic mode decomposition* (EDMD) [64]. In EDMD the best-approximation of the transfer operator within a finite-dimensional subspace is approximated based on trajectory data. This subspace is spanned by basis functions that can be freely chosen. If basis functions in the form of indicator functions that partition the state space are chosen, EDMD results in a so-called *Markov state model* (MSM) [62], which is especially popular in molecular dynamics. Similar to EDMD are *Ulam's method* [65] and *time-lagged independent component analysis* (TICA) [66]. In TICA, linear CVs that maximize autocorrelation are calculated from trajectory data, but it has been shown that it actually implicitly utilizes an approximation of the eigenvalues and eigenfunctions of the transfer operator.

Recently, more and more algorithms involving artificial neural networks have been proposed. For example, *VAMPnets* [67] are neural networks that represent the CVs and that are trained to maximize a score based on the agreement between the subspace spanned by the CVs and the subspace spanned by the dominant eigenfunctions of the transfer operator. (This score is motivated by the *variational approach for Markov processes* (VAMP) [68], which encompasses a similar theory to the one presented above.) Another approach is called *invariant subspaces of Koopman operators with artificial neural networks* (ISOKANN) [69], where again the CVs are represented by a neural network and are trained to agree with the subspace spanned by the dominant eigenfunctions, but using a type of power iteration that eliminates the non-dominant spectrum.

The neural network architecture called *autoencoder* is another promising method for the discovery of CVs. For instance, the *time-lagged autoencoder* (TLA) [46] learns an encoding of the system state x into a low-dimensional latent vector (i.e., a CV), and a decoding back from the latent vector to the original state space, such that the decoded state approximates $\mathbb{E}[\boldsymbol{x}(\tau) \mid \boldsymbol{x}(0) = x]$ as well as possible. The TLA can be seen as a nonlinear extension of TICA. Another example is the *SINDy autoencoder* [70] for deterministic dynamics, in which the encoder attempts to transform the original system state into reduced coordinates that allow a representation of the dynamics in the form of a simple ODE.

While many of the above-mentioned methods focus on specific use cases, for example metastable systems in molecular dynamics, the theory of *lumpability* and *decomposability* as presented in [58], and the closely related *transition manifold approach* [56, 57] offer a more general framework to discover CVs. Moreover, it allows the formulation of explicit dynamical conditions for finding good CVs that can be validated during the data-driven computation, which is an advantage over the methods utilizing black-box approximators like neural networks. The transition manifold approach will be discussed in more detail later in chapter 6, where a modification for dynamical systems on networks will be presented.

Collective variables for systems on networks. The most intuitive and also most popular choice of observable to either quantify the state of a system on a network on a macroscopic scale, or to even develop a reduced model in the space of that observable, is given by forming an average over the agents' states. For instance, in the Kuramoto model of coupled oscillators the mean over the complex phases $e^{i\theta_j}$ of oscillators j provides insight into the global synchronization status of the system, and a reduced model in the form of an ODE of this average can be formulated [71].

A similar concept is often applied to discrete-state systems. In the most simple case that agents have a state of either 0 or 1, the average over states corresponds to the *share* of state 1 in the system. This quantity is often of interest in the underlying application. For example, in the SIS model (cf. example 2.2) it corresponds to the percentage of infectious nodes. In statistical physics, these averages of states are often related to the amount of order shown by the system. For example, in the Ising model of ferromagnetism [72] the nodes' states represent their spin and their average is called the *magnetization* of the system. Hence, this average over states is also called the *order parameter* in this context[1]. In the more general case that a node's state is one of finitely many discrete states, $x_i \in \{1, \dots, M\}$, the average contains the shares of each discrete state in the system. For instance, in the SIR model (cf. example 2.2) the percentage of susceptible, infectious, and removed agents would be observed.

However, the simple average often does not contain enough information to formulate a reduced model that is able to replicate the behavior of the original system, i.e., it is not a good CV. (Although it will be investigated under which conditions the shares *are* good CVs for the voter model later in chapter 5.) Thus, many extensions to higher-dimensional CVs have been proposed. For example, instead of measuring the global average, one can partition the network into subnetworks and measure separate averages for each subnetwork, which is especially useful for clustered networks [1]. Another approach is to include higher order *network motifs* in the CV [8, 74, 75, 76, 77, 78]. A network motif is a subgraph that occurs frequently in the network, for example a triangle, and the CV counts the occurrences of this motif where the contained nodes have certain states, e.g., the number of triangles where two nodes have state m and one node has state n. The simplest network motif is just a single node, and the resulting CV produces the aforementioned *shares* of the discrete states in the system. Increasing the complexity of the network motif, one counts the number of connected *pairs* of nodes with certain states, for example the number of pairs where one node has state m and the other state n, or equivalently the number of m-n edges. Increasing complexity even further, one could count the number of triplets with certain states, and so on. A popular modification is to consider motifs that include the degree of nodes, e.g., instead of counting the number of nodes in state m, one counts the number of nodes of degree k in state m separately for all occurring degrees k. Techniques allowing to construct a reduced model that utilizes these counts of network motifs will be discussed in section 2.4.

A common theme in the literature is that the CVs, for which a reduced model is proposed, are simply guessed by investigating the properties of the underlying dynamical model and network structure. While many reduced models of high quality have been devised this way, the approach is very sensitive to changes of the network or dynamics, and requires a lot of effort and numerical verification. Motivated by the success of data-driven methods for algorithmically learning optimal CVs in the field of molecular dynamics, a method tailored specifically to discrete-state dynamics on networks will be presented in chapter 6.

Concentration Effects. Even though the microscopic dynamics describing the behavior of single agents is stochastic in many models, the macroscopic behavior of the complete

[1] This is not limited to physical models. For example, in a bounded confidence model the number of *active* edges between nodes that can influence each other can be seen as an order parameter [73].

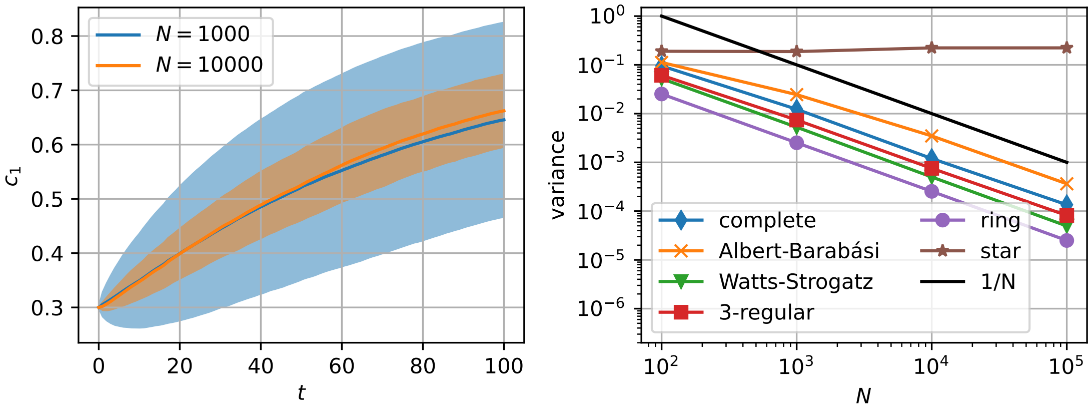


Figure 2.2.: Left: mean (solid line) ± standard deviation (shaded area) of the CNVM on an Albert–Barabási network, estimated from 1000 simulations with identical initial state. Right: mean ensemble variance, i.e., the time average of $\mathrm{Var}(\boldsymbol{c}_1(t))$, for different network topologies.

system consisting of many agents can sometimes be characterized by an aggregate observable that evolves (almost) deterministically. In the fashion of the classical law of large numbers, the many random interactions of agents may produce an emergent outcome that is no longer random – at least in the limit of infinitely many agents. This phenomenon that the law of the stochastic process given by an aggregate observable of the system concentrates around a deterministic trajectory as the number of agents increases will be called a *concentration effect*.

The continuous-time noisy voter model (CNVM), see section 2.2, shows concentration effects with respect to the shares of each discrete state for many underlying network structures. As depicted in figure 2.2, the stochastic process $\boldsymbol{c}_1(t)$, which describes the share of opinion 1 in the CNVM, concentrates around a deterministic trajectory for Albert–Barabási networks (see section 3.1.5 for details) of increasing size N. The variance[2] of trajectories tends to 0 at the rate N^{-1}, not only for Albert–Barabási networks but for many other network topologies as well. However, there are also networks for which $\boldsymbol{c}_1(t)$ does not concentrate. For example, on a star-shaped graph the variance of trajectories remains approximately constant as N increases. This phenomenon will be discussed in more detail later in section 5.4.

If concentration effects are observed with respect to some type of average over the nodes' states, the macroscopic deterministic dynamics that the stochastic process converges to can sometimes be derived with the help of *mean-field theory*, which will be explored in the next section.

[2] The "variance of trajectories" refers to mean ensemble variance, i.e., the average over time of $\mathrm{Var}(\boldsymbol{c}_1(t))$.

2.4. Techniques for model reduction

This section contains some popular techniques for the model reduction of dynamical systems on networks. First, the concepts of *mean-field theory* and *mean-field limits* are discussed in section 2.4.1. The class of *moment closure methods* is presented in section 2.4.2, where, amongst other things, a *pair approximation* for the continuous-time noisy voter model is derived. Finally, section 2.4.3 addresses the notion of *lumpability* of Markov chains, and, in particular, *automorphism-based lumping* for dynamical systems on networks.

2.4.1. Mean-field limits

In *mean-field theory* [9, 8] a system consisting of many interacting components is simplified by considering only an *average* component that is exposed to an *average* interaction with other components (i.e., a mean field), hence reducing a system of many agents to a system that can be expressed in terms of only one average agent. For discrete-state systems, where every agent i has a state $x_i \in \{1, \dots, M\}$, this average agent is often defined via the probability vector $c \in [0,1]^M$ that describes the *concentration* or *share* of each discrete state in the system, i.e., c_m is the percentage of agents with state m, or equivalently, the probability that a randomly chosen agent has state m. In this setting, the dynamics of the average agent c are often described as an ODE. For continuous-state systems on the other hand, the average agent is usually defined by a probability distribution over the state space, which describes the distribution of the states of all agents, or equivalently, the probability distribution of the state of a randomly chosen agent. (For finitely many agents, this distribution has the form of an empirical measure.) The dynamics of the average agent are then often defined in the form of a PDE of the associated probability density.

This extremely rough model reduction only works well for specific systems fulfilling certain homogeneity assumptions. Generally, the application of mean-field theory can only be successful if the interacting components of the system are sufficiently *indistinguishable* and *interchangeable* [79]. In the case of dynamical systems on networks, this implies that the network must have a homogeneous structure, i.e., no clustering and no communities, and that states of neighboring nodes must exhibit a low stochastic correlation [80]. For many dynamical models, such as the voter model, the latter is given if node degrees in the network are large because then the influence on a node's state by one of its many neighbors is rather small. Hence, mean-field theory can often be successfully applied to dynamical systems on large, dense, and homogeneous graphs, but fails to produce a reasonable model reduction for more complex, heterogeneous, or sparse network topologies.

Assuming that the above assumptions hold, it is in many cases straightforward to apply mean-field theory to generate a reduced model, which is why this approach is one of the most popular methods in the field. Consider for instance the continuous-time noisy voter model (CNVM) from section 2.2. To recap, in the CNVM an agent i transitions from its current state m to a different state n at the rate

$$r_{m,n} \frac{d_{i,n}(x)}{d_i} + \tilde{r}_{m,n}, \tag{2.21}$$

where $d_{i,n}(x)$ denotes the number of neighbors of node i with opinion n, d_i is the degree of node i, and $r_{m,n}, \tilde{r}_{m,n} \geq 0$ are model parameters[3]. As it is a discrete-state system, the average agent will be described by the shares $c \in [0,1]^M$, $\|c\|_1 = 1$. By applying $d_{i,n}(x) = \sum_j A_{i,j}\delta_{\{x_j=n\}}$, where A is the adjacency matrix, the mean-field that an agent i with opinion m is exposed to is given by

$$\begin{aligned}
&\mathbb{E}\Big[r_{m,n}\frac{1}{d_i}\sum_{j=1}^{N} A_{i,j}\delta_{\{\boldsymbol{x}_j=n\}} + \tilde{r}_{m,n}\Big|\boldsymbol{x}_i = m\Big] \\
&\approx r_{m,n}\frac{1}{d_i}\sum_{j=1}^{N} A_{i,j}\mathbb{E}[\delta_{\{\boldsymbol{x}_j=n\}}] + \tilde{r}_{m,n} \\
&\approx r_{m,n}\frac{1}{d_i}\sum_{j=1}^{N} A_{i,j}c_n + \tilde{r}_{m,n} \\
&= r_{m,n}c_n + \tilde{r}_{m,n}.
\end{aligned} \tag{2.22}$$

In the second line of the above equation the assumption of stochastic independence of the agents' states was applied to remove the condition in the expectation, and in the third line the interchangeability of agents was used to assume that all agents' states are identically distributed according to c. Hence, the *average agent* is in state m with probability c_m and changes to state n at the rate $r_{m,n}c_n + \tilde{r}_{m,n}$, which yields a "flow" from c_m to c_n of magnitude $c_m(r_{m,n}c_n + \tilde{r}_{m,n})$. Aggregating these flows for all $n \neq m$ yields the differential equation

$$\frac{\mathrm{d}}{\mathrm{d}t}c_m = \sum_{\substack{n=1\\n\neq m}}^{M} c_n(r_{n,m}c_m + \tilde{r}_{n,m}) - c_m(r_{m,n}c_n + \tilde{r}_{m,n}). \tag{2.23}$$

The resulting system of ODEs can be written in compact form and will be called the *mean-field equation* (MFE)

$$\frac{\mathrm{d}}{\mathrm{d}t}c = \sum_{m\neq n} c_m(r_{m,n}c_n + \tilde{r}_{m,n})(e_n - e_m), \tag{2.24}$$

where $e_n, e_m \in \mathbb{R}^M$ are the unit vectors and the sum is over all combinations of m and n. (In literature focused on well-mixed chemically reacting systems, the MFE is commonly called *reaction-rate equation* instead [40].) Although it was not difficult to derive this mean-field approximation for the CNVM, it is hard to assess its quality for given model parameters and underlying network.

As the assumptions for mean-field theory can never be truly fulfilled for finite size systems, e.g., states of neighboring nodes might always be somewhat stochastically dependent, the reduced model given by the mean-field approximation inevitably exhibits an error. If one however considers the *large population limit* $N \to \infty$, i.e., letting the number of agents, and in our case the network, grow infinitely large, the mean-field approximation can become exact. In this case the mean-field approximation is also called the *mean-field limit*, *thermodynamic limit*, or *hydrodynamic limit*. Section 5 discusses the mean-field limit for the CNVM in detail, addressing the question of exactness and quality of the approximation (2.24) as the number of nodes goes to infinity.

[3] The parameter α was set to 1 for convenience. For a general $\alpha \in \mathbb{R}$, an additional average over the degrees has to be included in the mean-field below.

An alternative approach to define the mean-field limit for discrete-state dynamics is often applied to systems on lattices and similar regular structures: instead of the global shares $c \in [0,1]^M$ of each state, the state distribution in the form of a density function p over the lattice is considered, i.e., $p(x) \in [0,1]^M$ represents the concentration of each discrete state at location x on the lattice. For example, given a one-dimensional lattice with N nodes, the position $x \in [0,1]$ refers to the node i with $\frac{i-1}{N} \leq x < \frac{i}{N}$, and $p(x)$ to the state distribution of this node i. It has been shown for some systems that, as the number of nodes goes to infinity, the time evolution of p follows a (S)PDE of heat equation or reaction-diffusion equation type, for instance a (stochastic) Fischer–Kolmogorov–Petrovsky–Piskunov equation [81, 82, 83, 84]. Furthermore, an integro-PDE can be derived for systems on certain dense graphs using Graphon theory [85], which will be briefly addressed in section 3.2. In this case, the argument $x \in [0,1]$ of the density function p does not have a spatial interpretation but can simply be seen as a generalization of the node index for an infinitely large network. As these approaches do not facilitate a dimension reduction in the fashion of collective variables (cf. section 2.3), but describe the system in an infinitely dimensional function space, they will not be considered in detail in this work.

Similarly, the mean-field limits of continuous-state dynamics like the bounded confidence model are not the subject of this work. As mentioned before, the average agent is often represented by a distribution over the state space. For certain deterministic models, the evolution of this distribution has been shown to evolve according to a PDE in the form of a continuity equation (or transport equation) [86, 79]. In the case of certain stochastic systems, for example if each agent's state evolves according to a SDE, the model converges to a McKean–Vlasov process in the mean-field limit, which is an example of *propagation of chaos* [87]. The probability distribution of the states then evolves according to the PDE given by the associated Fokker–Planck equation (Kolmogorov forward equation) [25, 88]. There is a significant overlap with techniques used in related problems in soft matter physics and statistical physics, for example *dynamic density functional theory* [89, 73].

2.4.2. Moment closure methods

When dealing with (infinitely) large systems of coupled ODEs that describe the evolution of so-called *moments*, i.e., statistically relevant quantities related to the underlying system, *moment closure methods* [90] can be used to obtain a smaller and closed set of equations. Typically, the evolution of *lower-order* moments depends on *higher-order* moments. By expressing this dependence only in terms of the lower-order moments, a closure can be achieved. Hence, the application of moment closure methods usually entails an approximation error whose magnitude depends on the quality of the closure, i.e., on how well the higher-order moments could be expressed by the lower-order moments.

In the context of (discrete) dynamical systems on networks, moment closure methods are often applied to the evolution equations of the frequency of *network motifs* [8, 39, 74], e.g., the frequency of single nodes in a certain state, the frequency of linked pairs (neighbors) in certain states, or the frequency of triplets in certain states. These equations are usually hierarchical, i.e., the equation for single nodes contains the frequency for pairs, the equation for pairs contains the frequency of triplets, and so on. Hence, a closure of the equations has to be performed to eliminate dependency on higher-order motifs.

Moment closure for the voter model. In the following, this approach is demonstrated for a simple example: the continuous-time noisy voter model (CNVM) from section 2.2, with $M = 2$ discrete states and parameters

$$r = \begin{pmatrix} - & \lambda \\ 0 & - \end{pmatrix}, \quad \tilde{r} = \begin{pmatrix} - & 0 \\ 0 & - \end{pmatrix}, \quad \alpha = 1. \tag{2.25}$$

It can be seen as a SI-model without recovering, i.e., after a node has become infected it will stay infected forever. Let $\boldsymbol{x}_i(t) \in \{0, 1\}$ denote the stochastic process describing the state of the i-th node over time, where 0 represents susceptible and 1 infected. Per definition of the model, the rate at which a susceptible node i becomes infected is given by $\lambda \frac{1}{d_i} \sum_j A_{i,j} x_j$, where A is the adjacency matrix of the underlying network. As the rate for an infected node to become susceptible again is zero, the expected value of the state of node i changes at the rate

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} \mathbb{E}[\boldsymbol{x}_i] &= \mathbb{P}(\boldsymbol{x}_i = 0)\ \mathbb{E}\Big[\lambda \frac{1}{d_i} \sum_j A_{i,j} \boldsymbol{x}_j \Big| \boldsymbol{x}_i = 0\Big] \\ &= \mathbb{P}(\boldsymbol{x}_i = 0)\ \lambda \frac{1}{d_i} \sum_j A_{i,j} \mathbb{E}[\boldsymbol{x}_j | \boldsymbol{x}_i = 0]. \end{aligned} \tag{2.26}$$

Inserting the equality

$$\mathbb{P}(\boldsymbol{x}_i = 0)\ \mathbb{E}[\boldsymbol{x}_j | \boldsymbol{x}_{\boldsymbol{i}} = 0] = \mathbb{P}(\boldsymbol{x}_i = 0, \boldsymbol{x}_j = 1) = \mathbb{E}[(1 - \boldsymbol{x}_i)\boldsymbol{x}_j] \tag{2.27}$$

yields the evolution equation for the first moment

$$\frac{\mathrm{d}}{\mathrm{d}t} \mathbb{E}[\boldsymbol{x}_i] = \lambda \frac{1}{d_i} \sum_j A_{i,j} \mathbb{E}[(1 - \boldsymbol{x}_i)\boldsymbol{x}_j], \tag{2.28}$$

which depends on the higher order moment $\mathbb{E}[(1 - \boldsymbol{x}_i)\boldsymbol{x}_j]$. Hence, the probability of single nodes being infected (lower-order motif) depends on the probabilities of neighboring pairs consisting of one infected and one susceptible node (higher-order motif). A closure can for example be achieved by assuming that all nodes are stochastically independent, from which the closed system of equations

$$\frac{\mathrm{d}}{\mathrm{d}t} \mathbb{E}[\boldsymbol{x}_i] = \lambda \frac{1}{d_i} \sum_j A_{i,j} (1 - \mathbb{E}[\boldsymbol{x}_i]) \mathbb{E}[\boldsymbol{x}_j], \quad i = 1, \dots, N \tag{2.29}$$

follows. Note that there is a large number N of equations; one for each node i. Further simplifications can be conducted if more assumptions about the system are made. For instance, assuming that all nodes are interchangeable and behave (on average) identically, $\mathbb{E}[\boldsymbol{x}_i]$ can be replaced by the share $c \in [0, 1]$ of infected nodes for all i, resulting in the *mean-field equation* (cf. section 2.4.1 and eq. (2.24))

$$\frac{\mathrm{d}}{\mathrm{d}t} c = \lambda \frac{1}{d_i} \sum_j A_{i,j} (1 - c) c = \lambda (1 - c) c. \tag{2.30}$$

Hence, the reduced model obtained by closing after the first moment is often called the *mean-field approximation.*

However, the assumption that all nodes are stochastically independent is very strong and thus can potentially introduce a large error if it is violated substantially. Therefore, one might be motivated to investigate the higher-order equations instead of already closing at the first moment to improve the accuracy of the approximation. The second order moment $\mathbb{E}[(1-\boldsymbol{x}_i)\boldsymbol{x}_j]$ experiences a positive change if $(\boldsymbol{x}_i,\boldsymbol{x}_j)=(0,0)$ and then node j transitions to state 1, and a negative change if $(\boldsymbol{x}_i,\boldsymbol{x}_j)=(0,1)$ and then node i transitions to state 1. Thus, the rate of change of the second moment is given by

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}[(1-\boldsymbol{x}_i)\boldsymbol{x}_j] = \mathbb{P}(\boldsymbol{x}_i=0,\boldsymbol{x}_j=0)\ \mathbb{E}\Big[\lambda\frac{1}{d_j}\sum_{k=1}^{N}A_{j,k}\boldsymbol{x}_k\Big|\boldsymbol{x}_i=0,\boldsymbol{x}_j=0\Big]\\ -\mathbb{P}(\boldsymbol{x}_i=0,\boldsymbol{x}_j=1)\ \mathbb{E}\Big[\lambda\frac{1}{d_i}\sum_{k=1}^{N}A_{i,k}\boldsymbol{x}_k\Big|\boldsymbol{x}_i=0,\boldsymbol{x}_j=1\Big].\end{aligned} \tag{2.31}$$

Applying the linearity of the expected value and considering the terms where $k=i,j$ separately yields

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}[(1-\boldsymbol{x}_i)\boldsymbol{x}_j] = \mathbb{P}(\boldsymbol{x}_i=0,\boldsymbol{x}_j=0)\ \lambda\frac{1}{d_j}\sum_{\substack{k=1\\k\neq i}}^{N}A_{j,k}\mathbb{E}[\boldsymbol{x}_k|\boldsymbol{x}_i=0,\boldsymbol{x}_j=0]\\ -\mathbb{P}(\boldsymbol{x}_i=0,\boldsymbol{x}_j=1)\ \lambda\frac{1}{d_i}\sum_{\substack{k=1\\k\neq j}}^{N}A_{i,k}\mathbb{E}[\boldsymbol{x}_k|\boldsymbol{x}_i=0,\boldsymbol{x}_j=1]\\ -\mathbb{P}(\boldsymbol{x}_i=0,\boldsymbol{x}_j=1)\ \lambda\frac{1}{d_i}\mathbb{E}[\boldsymbol{x}_j|\boldsymbol{x}_i=0,\boldsymbol{x}_j=1].\end{aligned} \tag{2.32}$$

By the definition of the conditional expectation (analogously to (2.27)), the evolution equation of the second moment follows:

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}[(1-\boldsymbol{x}_i)\boldsymbol{x}_j] = \lambda\frac{1}{d_j}\sum_{\substack{k=1\\k\neq i}}^{N}A_{j,k}\mathbb{E}[\boldsymbol{x}_k(1-\boldsymbol{x}_i)(1-\boldsymbol{x}_j)]\\ -\lambda\frac{1}{d_i}\sum_{\substack{k=1\\k\neq j}}^{N}A_{i,k}\mathbb{E}[\boldsymbol{x}_k(1-\boldsymbol{x}_i)\boldsymbol{x}_j]\\ -\lambda\frac{1}{d_i}\mathbb{E}[(1-\boldsymbol{x}_i)\boldsymbol{x}_j].\end{aligned} \tag{2.33}$$

After inserting $\mathbb{E}[\boldsymbol{x}_k(1-\boldsymbol{x}_i)(1-\boldsymbol{x}_j)] = \mathbb{E}[\boldsymbol{x}_k(1-\boldsymbol{x}_j)] - \mathbb{E}[\boldsymbol{x}_k\boldsymbol{x}_i(1-\boldsymbol{x}_j)]$, the only occurring third order moments are of the form $\mathbb{E}[\boldsymbol{x}_k(1-\boldsymbol{x}_i)\boldsymbol{x}_j]$, where node i is connected to node j and to node k. They represent the probability that a connected triplet $k-i-j$ has states $1-0-1$. To close these equations, one can for example assume that nodes k and j of such a triplet are always stochastically independent. By the definition of conditional probability this yields

$$\begin{aligned}\mathbb{E}[\boldsymbol{x}_k(1-\boldsymbol{x}_i)\boldsymbol{x}_j] &= \mathbb{P}(\boldsymbol{x}_k=1,\boldsymbol{x}_i=0,\boldsymbol{x}_j=1)\\ &= \mathbb{P}(\boldsymbol{x}_k=1|\boldsymbol{x}_i=0,\boldsymbol{x}_j=1)\ \mathbb{P}(\boldsymbol{x}_i=0,\boldsymbol{x}_j=1)\\ &= \frac{\mathbb{P}(\boldsymbol{x}_k=1,\boldsymbol{x}_i=0)}{\mathbb{P}(\boldsymbol{x}_i=0)}\mathbb{P}(\boldsymbol{x}_i=0,\boldsymbol{x}_j=1)\\ &= \frac{\mathbb{E}[\boldsymbol{x}_k(1-\boldsymbol{x}_i)]}{1-\mathbb{E}[\boldsymbol{x}_i]}\mathbb{E}[(1-\boldsymbol{x}_i)\boldsymbol{x}_j],\end{aligned} \tag{2.34}$$

which only consists of first and second order moments. When inserting this into (2.33), the so-called *pair approximation* [8, 78, 91] is obtained

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}[(1-\boldsymbol{x}_i)\boldsymbol{x}_j] = \lambda\frac{1}{d_j}\sum_{\substack{k=1\\k\neq i}}^{N} A_{j,k}\Big(\mathbb{E}[\boldsymbol{x}_k(1-\boldsymbol{x}_j)] - \frac{\mathbb{E}[\boldsymbol{x}_k(1-\boldsymbol{x}_j)]}{1-\mathbb{E}[\boldsymbol{x}_j]}\mathbb{E}[(1-\boldsymbol{x}_j)\boldsymbol{x}_i]\Big)\\ -\lambda\frac{1}{d_i}\sum_{\substack{k=1\\k\neq j}}^{N} A_{i,k}\Big(\frac{\mathbb{E}[\boldsymbol{x}_k(1-\boldsymbol{x}_i)]}{1-\mathbb{E}[\boldsymbol{x}_i]}\mathbb{E}[(1-\boldsymbol{x}_i)\boldsymbol{x}_j]\Big)\\ -\lambda\frac{1}{d_i}\mathbb{E}[(1-\boldsymbol{x}_i)\boldsymbol{x}_j].\end{aligned} \tag{2.35}$$

Together with the first order equations (2.28), the pair approximation contains a closed system of $N+2E$ equations, where N is the number of nodes and E the number of edges of the graph. The pair approximation generally has a lower error than the mean-field approximation because it utilizes fewer assumptions about the system: While the mean-field approximation requires pairwise stochastic independence of all nodes, the pair approximation only requires that for all triplets the two outer nodes are stochastically independent.[4]

If again the interchangeability of agents is assumed, all $\mathbb{E}[(1-\boldsymbol{x}_i)\boldsymbol{x}_j]$ can be replaced by the same quantity $s = 0.5E_\mathrm{a}/E$, where E_a denotes the number of *active* edges, i.e., edges between an infected and a susceptible node. Thus, E_a/E is the probability that a randomly chosen edge $\{i,j\}$ is between an infected and a susceptible node, and when halved it corresponds to the probability $\mathbb{E}[(1-\boldsymbol{x}_i)\boldsymbol{x}_j]$ that i is susceptible and j infected. Inserting s into (2.35) yields

$$\frac{\mathrm{d}}{\mathrm{d}t}s = \lambda\frac{d_j-1}{d_j}\left(s-\frac{s^2}{1-c}\right) - \lambda\frac{d_i-1}{d_i}\frac{s^2}{1-c} - \lambda\frac{1}{d_i}s, \tag{2.36}$$

which is not closed due to the dependence on the degrees d_i and d_j of nodes i and j. In the mean-field equation (2.30), this was not an issue because the degrees canceled out[5]. However, as the interchangeability of nodes has already been assumed, it is plausible to simply replace all degrees with the average degree d, leading to the pair approximation

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}c &= \lambda s\\ \frac{\mathrm{d}}{\mathrm{d}t}s &= \lambda\frac{d-1}{d}\left(s-2\frac{s^2}{1-c}\right) - \lambda\frac{1}{d}s.\end{aligned} \tag{2.37}$$

[4] If the triplet is a clique, this condition still implies pairwise stochastic independence.

[5] This is not a general property of the mean-field approximation but a consequence of the chosen dynamics. If $\alpha \neq 1$ had been chosen instead in the CNVM, the degrees would not have canceled out.

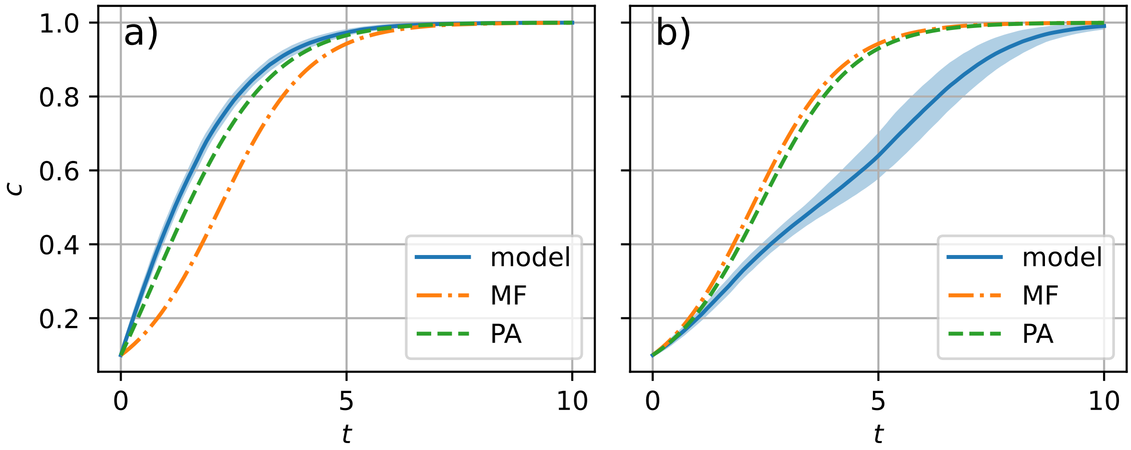


Figure 2.3.: Share of infected nodes c over time for the SI model (2.25), mean-field approximation (MF) (2.30), and pair approximation (PA) (2.37). a): Albert–Barabási network, $N = 1000$. b): network with two sparsely connected communities, $N = 1000$.

The quality of the mean-field approximation and the pair approximation is investigated in figure 2.3 for an Albert–Barabási network (see section 3.1.5 for more details) and for a network composed of two sparsely interconnected communities. For the scale-free Albert–Barabási network, the 10% of nodes with the largest degrees are initially infected, and for the clustered network the initially infected 10% all belong to the same community. In the case of the Albert–Barabási network, the mean-field approximation exhibits a significant error as it can not resolve the large number of active edges due to the initial state and hence massively underestimates the rate of infection. The pair approximation on the other hand has access to the share of active edges, resulting in a substantially smaller approximation error. However, both the mean-field and the pair approximation fail in the case of the clustered network, see figure 2.3 b). As the infection first spreads in one of the clusters before jumping to the other one, both approximations overestimate the global rate of infection. For such clustered networks it is crucial to choose collective variables that are able to resolve this cluster structure, which the counts of network motifs typically investigated in moment closure methods can not do. In some cases it is sufficient to measure network motifs, e.g., the share of infected nodes, separately for each cluster, which is discussed in more detail in section 5.2.3. However, for complicated network structures it might not be apparent which choice of collective variables is appropriate. For such cases a data-driven algorithm that is able to automatically derive optimal collective variables is presented in chapter 6.

Heterogeneous approximations and the master equation. Previously, the assumption that all nodes are interchangeable was used to reduce the number of equations to one for the mean-field approximation and to two for the pair approximation. Hence, these types of approximations are called *homogeneous*. They typically involve the substitution of the different node degrees occurring in the equations with the mean node degree. This can result in large errors especially when the underlying degree distribution is very heterogeneous and contains degrees of vastly different sizes. In such cases it is common to only assume interchangeability of nodes of the same degree, which results in a larger number of equations that resolve the possibly different behavior of nodes with different degrees. The obtained approximations are thus called *heterogeneous* [75, 76, 77]. For instance, instead of describing the share c of infected nodes as in the homogeneous mean-field approximation (2.30), the heterogeneous mean-field approximation would describe the share c_k of nodes that have degree k and are infected, for each degree k occurring in the graph. As a consequence, the number of equations in heterogeneous approximations scales with the number of different degrees that can be found in the network.

A further refinement of these heterogeneous approximations is given by the *approximate master equation* (AME) [92], which in addition to the node degree also resolves the number of infected neighbors of nodes in order to produce an even more accurate approximation. The AME provides evolution equations for the quantities $i_{k,m}$ and $s_{k,m}$ denoting the share of infected and susceptible nodes that have degree k and exactly m neighbors that are infected, $m \in \{0, \dots, k\}$. It has been shown that the AME is significantly more accurate than simple mean-field or pair approximations. In fact, they can be obtained from the AME by aggregating appropriate quantities [93, 75]. However, the AME still involves assumptions about stochastic independence and interchangeability of nodes such that certain (local) structural correlations in the network are ignored, inducing an approximation error.

2.4.3. Lumpability

In the context of continuous-time Markov chains, like the continuous-time noisy voter model (CNVM) presented in section 2.2, the notion of *lumpability* [94] is of significant importance when searching for model reductions.

Definition 2.3. A continuous-time Markov chain with generator matrix Q is called *lumpable* with respect to a partition $\{\mathcal{Y}_1, \dots, \mathcal{Y}_L\}$ of its finite state space if for every pair of sets $(\mathcal{Y}_k, \mathcal{Y}_l)$, $k \neq l$, it follows

$$\forall x, y \in \mathcal{Y}_k : \sum_{w \in \mathcal{Y}_l} Q_{x,w} = \sum_{w \in \mathcal{Y}_l} Q_{y,w}. \tag{2.38}$$

Thus, every starting state in $\mathcal{Y}_k$ has the same rate of transitioning to somewhere in $\mathcal{Y}_l$.

The significance of the above definition is explained by the fact that a lumpable Markov chain allows the construction of an associated aggregated process that is itself Markovian. More precisely, let $\boldsymbol{x}(t)$ denote the state of the continuous-time Markov chain. Given the partition $\{\mathcal{Y}_1, \dots, \mathcal{Y}_L\}$, the aggregated process $\boldsymbol{y}(t) \in \{1, \dots, L\}$ is defined by tracking in which set of the partition $\boldsymbol{x}(t)$ is in, i.e.,

$$\boldsymbol{y}(t) = k \ :\Leftrightarrow \ \boldsymbol{x}(t) \in \mathcal{Y}_k. \tag{2.39}$$

Theorem 2.4 ([94])**.** *Given a partition $\{\mathcal{Y}_1, \dots, \mathcal{Y}_L\}$ of the state space of $\boldsymbol{x}(t)$, the associated aggregated process $\boldsymbol{y}(t)$ is a continuous-time Markov chain if and only if $\boldsymbol{x}(t)$ is lumpable with respected to $\{\mathcal{Y}_1, \dots, \mathcal{Y}_L\}$. In that case, the rate matrix of $\boldsymbol{y}(t)$ is given by $\tilde{Q}_{k,l} = \sum_{w \in \mathcal{Y}_l} Q_{x,w}$, where $x \in \mathcal{Y}_k$.*

Thus, in the case of a lumpable Markov chain, the aggregated process can be seen as a reduced model, and the map assigning to a state x the index k such that $x \in \mathcal{Y}_k$ can be interpreted as the corresponding collective variable.

Note that the lumping together of states is always a trade-off between size reduction and loss of information. For example, every Markov chain is lumpable with respect to the partition consisting of only one set: the whole state space. However, the associated aggregated process contains no information. On the other hand, every Markov chain is also lumpable with respect to the partition containing only sets with cardinality one, one for each state. However, the associated aggregated process offers no reduction in size. As a consequence, lumpability itself is not a sufficient condition for the existence of a meaningful reduced model, and the associated collective variable is not automatically *good*, as defined in section 2.3. A common strategy to circumvent this issue is to propose a partition $\mathcal{P}$ that contains the information one is interested in, and then to find the coarsest lumpable partition that is at least as fine as $\mathcal{P}$. Hence, the associated aggregated process has minimal size while still containing the desired information and being Markovian.

However, for discrete-state dynamical systems on networks such as the CNVM the task of finding lumpable partitions seems practically impossible. If each node has one of M discrete states, the state space of the process contains M^N elements, where N denotes the number of nodes. Thus, the number of partitions of the state space is more than[6] $\exp(M^N)$, which is astronomically large even for relatively small M and N. For instance, given $M = 2$ and $N = 10$ there are substantially more than 10^{400} possible partitions. Hence, a brute-force search for lumpable partitions is computationally infeasible.

Fortunately, the effective number of relevant partitions can be significantly reduced in some cases by exploiting symmetries in the underlying network, which is discussed in the following paragraph.

Automorphism-based lumping In [95] it was observed that graph automorphisms play a central role when applying the concept of lumpability to discrete-state systems on networks.

Definition 2.5. A bijection $f : V \to V$ is called an *automorphism* on the simple graph $G = (V, E)$ if for all $i, j \in V$:

$$\{i, j\} \in E \;\Leftrightarrow\; \{f(i), f(j)\} \in E. \tag{2.40}$$

The set of all automorphisms $\mathrm{Aut}(G)$ of the graph G forms a group with respect to composition and is a subgroup of the symmetric group $\mathrm{Sym}(V)$. The action of $\mathrm{Aut}(G)$ on the state space $[M]^N$ of the model is defined by

$$f \cdot x = y \;:\Leftrightarrow\; \forall i \in [N] : \; x_{f(i)} = y_i, \tag{2.41}$$

where $x, y \in [M]^N$ and $f \in \mathrm{Aut}(G)$.

The action defined above induces an equivalence relation on $[M]^N$ by associating states that can be converted into one another by applying an automorphism, i.e.,

$$x \sim y \;:\Leftrightarrow\; \exists f \in \mathrm{Aut}(G) : f \cdot x = y. \tag{2.42}$$

[6] The actual number of partitions is given by the Bell numbers, which even grow faster than exponentially.

Given the assumption that the dynamical model is homogeneous in the sense that the rate at which a node transitions from one state to another is calculated the same way for every node and depends only on the states of neighboring nodes, the following theorem can be shown.

Theorem 2.6 ([95]). *The continuous-time Markov chain $\boldsymbol{x}(t)$ is lumpable with respect to the partition given by the quotient space $[M]^N/{\sim}$. Moreover, this partition is at least as fine as the partition that merges all states with identical node state shares.*

The above theorem does not only address the existence of a lumpable partition but offers an explicit construction. For example, the equivalence class of a state $x \in [M]^N$, which is a member of the partition, is given by its orbit $\{f \cdot x \mid f \in \operatorname{Aut}(G)\}$. In addition, it is guaranteed that the partition contains at least the information of state counts, i.e., two states x and y with differing shares of each node state will never belong to the same equivalence class. For instance, in an SI model the number of infected and susceptible nodes can always be inferred from the aggregated process, which is an important and meaningful observable for most applications.

Although the lumpable partition $[M]^N/{\sim}$ is explicitly known, it is intractable to find it for all but the simplest graphs. Two such simple examples are discussed below. For most other graphs however, only a numerical solution is practical.

Example 2.7. For the sake of simplicity, assume that $M = 2$ in the following examples. One node state will be referred to as infected and the other as susceptible.

a) In the case of a complete graph, it is easy to see that every permutation is an automorphism, i.e., $\operatorname{Aut}(G) = \operatorname{Sym}([N])$. Hence, a state $x \in \{0,1\}^N$ can be transformed into every other state that has the same number of infected nodes by an automorphism, and the lumpable partition consists of $N+1$ sets $\{\mathcal{Y}_0, \dots, \mathcal{Y}_N\}$, where $\mathcal{Y}_k$ contains all the states x with k infected nodes.

b) For a star graph consisting of a central node and $N-1$ outer nodes that are only connected to the central node, the automorphism group consists of all permutations that do not affect the central node, i.e., $\operatorname{Aut}(G) \simeq \operatorname{Sym}([N-1])$. Hence, two states are equivalent $x \sim y$ if the central node has the same state and the outer nodes have the same count of infected nodes. The resulting lumpable partition therefore consists of $2N$ sets, N for the case that the central node is infected and N for the case that it is susceptible.

Discussion on lumpability. Although the approach of automorphism-based lumping seems promising, there are several problems limiting its usefulness and restricting its suitability for finding collective variables and reduced models. First of all, most networks do not exhibit many symmetries, i.e., the automorphism group $\operatorname{Aut}(G)$ is small, which results in only an insignificant reduction in size of the aggregated process. It is not unusual for large networks to even have no symmetries, $\operatorname{Aut}(G) = \{\mathrm{id}\}$, so that this approach can not be applied at all. For instance, Erdős–Rényi random graphs (see section 3.1.1 for more details) have no symmetries asymptotically almost surely [96]. The cause of these issues is the rigidity of the underlying concept of lumpability: per definition the aggregated process is an *exact* projection of the original process. An approximation error of the reduced dynamics is not allowed. Thus, lumpability requires much more than what the notion of collective variables is about, i.e., trading off a small approximation error in the reduced model for a large reduction in size and complexity in an intelligent way.

To address this rigidity of lumpability some relaxations have been studied in the literature, see for example [97, 98, 99]. The most noteworthy extension in the given context is presented in [96], where the approach of automorphism-based lumping is relaxed to only require *local* symmetries of the graph. While this extends the number of cases where the approach can be applied effectively, the underlying combinatorial and computational issues, which were also briefly discussed above, still make employing the lumpability approach exceptionally hard or even infeasible for larger graphs.

To circumvent these combinatorial issues one can instead work with a continuous version of lumpability, for instance as defined in [58] for continuous-state Markov processes. In this setting, a Markov process is called lumpable if there exists a collective variable φ such that every transition density function p_x^t (see section 2.3 for definition) depends essentially only on the value $\varphi(x)$ and not on the initial state x itself[7]. In a sense, the level sets of the collective variable take the role of the partition in the discrete version of lumpability, i.e., for initial states x and y with $\varphi(x) \approx \varphi(y)$ a lumpable system behaves (almost) identically, $p_x^t \approx p_y^t$. Note that this definition of lumpability is very similar to the intuitive conditions on a *good* collective variable that were introduced in section 2.3. The concept of *transition manifolds* is closely related to lumpability [58] and will be discussed in more detail in chapter 6, where a data-driven algorithm to find good collective variables for dynamical systems on networks will be presented.

[7] For intermediate lag-times t.

3. Random graphs

A crucial part of modeling real-world phenomena as dynamical systems on networks is an appropriate choice of the underlying graph structure. In some areas of application highly resolved network data is available. Typical examples are online social media, transport networks (e.g. flights), and power grids. However, in many applications the acquisition of such detailed network data is impractical or even impossible. For instance, in large scale opinion dynamics it is challenging to measure the relation between every pair of individuals. If reliable network data is not available, *random graphs* can be used to construct surrogate networks that mimic the properties of the corresponding real-world network.

The measurement and analysis of real-world social networks is the subject of *social network analysis* [100]. Numerous studies (see [101, 102, 103] for an overview) indicate that real (social) networks often have the following properties:

1. **Heavy-tailed degree distribution.** The degree distribution follows a power law, i.e., $f(d) \propto d^{-\alpha}$, where $f(d)$ is the fraction of nodes with degree d and $\alpha > 0$. In such a graph, there are nodes with very large degree, which are called *hubs*. However, the vast majority of nodes have rather small degree. Typically, the exponent takes values $2 \leq \alpha \leq 3$, in which case the network is also called *scale-free* [103]. (Recent work [104] argues that a log-normal distribution is often better suited to represent real-world networks than a power law.)

2. **Communities and clustering.** Nodes can be grouped into communities such that they are much more densely connected to nodes within their community than to other communities, i.e., the graph exhibits a large *modularity* [105]. Moreover, the *clustering coefficient* [106], which measures how close on average the neighborhood of a node is to be a clique (complete graph), is large in real networks. Intuitively, a large clustering coefficient means that if two nodes share the same neighbor, they are also likely connected.

3. **Small-world property.** The average path length and the diameter (i.e., the longest of the shortest paths between any two nodes) are comparatively small. If at the same time the clustering coefficient is not small, the network is said to exhibit the *small-world property* [106].

Moreover, especially social networks tend to exhibit a large *assortativity* due to the homophiliy of nodes, i.e., edges are more likely to form between similar nodes (e.g., nodes with similar degree) [23].

Using different random graph models it is possible to easily construct artificial networks that emulate one or several of the above characteristics of real-world networks [107]. This thesis focuses mainly on simple graphs, i.e., graphs with undirected and unweighted edges between exactly two nodes.

Definition 3.1 (Random graph)**.** Let $\mathcal{G}_N$ denote the set of all simple graphs $G = (V, E)$ with node set $V = [N]$. There are $|\mathcal{G}_N| = 2^{\binom{N}{2}}$ such simple graphs. Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space. A random variable $\boldsymbol{G} : \Omega \to \mathcal{G}_N$ is called a *random graph*.

A graph property $\mathcal{P}_N \subset \mathcal{G}_N$ is fulfilled with probability $\mathbb{P}(\boldsymbol{G} \in \mathcal{P}_N)$. For example, if $\mathcal{P}_N = \{G \in \mathcal{G}_N \mid G \text{ is connected}\}$, then $\mathbb{P}(\boldsymbol{G} \in \mathcal{P}_N)$ specifies the probability that $\boldsymbol{G}$ is connected. It is interesting to examine if certain graph properties hold as the size N of the random graph increases.

Definition 3.2. Let $(\boldsymbol{G}_\ell)_{\ell \in \mathbb{N}}$ be a sequence of random graphs $\boldsymbol{G}_\ell : \Omega \to \mathcal{G}_{N_\ell}$, such that $\lim_{\ell \to \infty} N_\ell = \infty$. Let $\mathcal{P} = (\mathcal{P}_{N_\ell})_{\ell \in \mathbb{N}}$ be a graph property. The sequence of random graphs fulfills $\mathcal{P}$ *asymptotically almost surely* (a.a.s.) if

$$\lim_{\ell \to \infty} \mathbb{P}(\boldsymbol{G}_\ell \in \mathcal{P}_{N_\ell}) = 1. \tag{3.1}$$

The remaining part of this chapter is organized as follows. In section 3.1 the most commonly used random graph models and their properties are discussed. Section 3.2 is concerned with graphons, which can be used to represent certain random graphs as functions on the unit square. Finally, the property of a random graph to be invariant under graph isomorphism is discussed in section 3.3.

3.1. Commonly used models

In this section some of the most commonly used random graph models are discussed, i.e., Erdős–Rényi random graphs, the stochastic block model, the Watts–Strogatz small world model, the Albert–Barabási model, and the configuration model.

3.1.1. Erdős–Rényi random graphs

The *Erdős–Rényi (ER) random graph* or *binomial random graph* $\boldsymbol{G}_{N,p}$ is the random graph with N nodes where each edge is present with probability $p \in [0, 1]$, independently of the other edges [108]. Hence, for all $G = (V, E) \in \mathcal{G}_N$ the ER random graph satisfies

$$\mathbb{P}(\boldsymbol{G}_{N,p} = G) = p^{|E|}(1 - p)^{\binom{N}{2} - |E|}. \tag{3.2}$$

The degree distribution of $\boldsymbol{G}_{N,p}$ is binomial, i.e., if the random variable $\boldsymbol{d}_i$ denotes the degree of an arbitrary node i, it follows that

$$\mathbb{P}(\boldsymbol{d}_i = k) = \binom{N-1}{k} p^k (1 - p)^{N-1-k}. \tag{3.3}$$

This results in homogeneous networks where most nodes have a degree close to the expected degree $(N - 1)p$, which is substantially different from the heavy-tailed degree distributions of real-world networks. The concentration of node degrees is addressed in the following lemma, which follows directly from the Chernoff bound (cf. Lemma A.1).

Lemma 3.3. *Let $\boldsymbol{G}_{N,p}$ denote the ER random graph and the random variable $\boldsymbol{d}_i$ the degree of an arbitrary node i. Then for all $\varepsilon > 0$ it follows that*

$$\mathbb{P}\big(|\boldsymbol{d}_i - \mu| \geq \varepsilon\mu\big) \leq 2 \exp\left(-\frac{\varepsilon^2 \mu}{3}\right), \tag{3.4}$$

where $\mu \coloneqq \mathbb{E}[\boldsymbol{d}_i] = (N - 1)p$.

Many properties of $\boldsymbol{G}_{N,p}$ change drastically depending on p. It is implicitly assumed that $p = p(N)$ is a function of the number of nodes N in order to discuss the asymptotic behavior. Some interesting properties are summarized below (see [109] for proofs):

1. If $p \ll \frac{1}{N}$, the random graph is a.a.s. not connected. Each connected component consists of a vanishingly small share of nodes.
2. If $\frac{1}{N} \ll p \ll \frac{\log N}{N}$, the random graph is still a.a.s. not connected. There exists a unique giant component of size $\mathcal{O}(N)$, i.e., it contains a non-vanishing fraction of nodes. All other connected components have sizes of at most $\mathcal{O}(\log N)$.
3. If $p \gg \frac{\log N}{N}$, the random graph is a.a.s. connected. The diameter is approximately $\operatorname{diam}(\boldsymbol{G}_{N,p}) \approx \log N / \log Np$. For a constant p, the diameter equals 2 a.a.s.
4. If $Np \to \infty$, all degrees are asymptotically equal, in the sense that for all $\varepsilon > 0$ every degree $\boldsymbol{d_i}$ is in the interval $[Np(1-\varepsilon), Np(1+\varepsilon)]$ with high probability (see Lemma 3.3).
5. The expected local clustering coefficient and average clustering coefficient of $\boldsymbol{G}_{N,p}$ is p. Thus, the resulting networks do not show clustering, as the coefficient only reflects the overall edge density p.

Due to the outlined properties of the ER random graph, it is not suitable to generate realistic (social) networks. While a small diameter can be realized by choosing a large p, the degree distribution of $\boldsymbol{G}_{N,p}$ remains substantially different from many real-world networks as it lacks a community structure (low modularity) and local clustering. However, the ER random graph is still one of the most used and studied random graph models because of its simplicity and the stochastic independence of edges.

3.1.2. The stochastic block model

The stochastic block model [110] is a generalization of the Erdős–Rényi random graph that makes including a community structure possible. The set of nodes $V = [N]$ is partitioned into K disjoint subsets $V_1, \ldots, V_K$, which will form the communities. Given a symmetric matrix $P \in [0,1]^{K \times K}$ of edge probabilities, a node $i \in V_k$ is connected with a node $j \in V_{k^*}$ with probability P_{k,k^*}. The choice $K = 1$ yields the standard ER random graph.

To generate a community structure in the sense of a large modularity, the intra-probabilities $P_{k,k}$ are typically chosen larger than the inter-probabilities P_{k,k^*}, $k \neq k^*$. The remaining graph properties can largely be transferred from the ER random graph, see section 3.1.1.

3.1.3. Random graphs with given expected degrees

Another generalization of the Erdős–Rényi random graph model is the *random graph with given expected degrees*. After specifying the expected degree $w_i \geq 0$ for each node $i = 1, \ldots, N$, an edge between nodes i and j is inserted with probability $p_{i,j} := w_i w_j / \sum_{k=1}^{N} w_k$, independently of other edges. The condition $\max_i w_i^2 \leq \sum_i w_i$ on the expected degrees ensures that $p_{i,j} \in [0,1]$. As a result, the expected degree of node i is indeed w_i. The ER random graph $\boldsymbol{G}_{N,p}$ is a special case of this procedure by choosing $w_i = Np$ for all i. Many properties of the ER random graph can be extended to this model. For instance, if the average degree $d \coloneqq \sum_{i=1}^{N} w_i/N$ is strictly larger than 1, there exists a unique giant component of $\mathcal{O}(N)$ size [111]. Moreover, under certain conditions on the expected degrees, the diameter is approximately $\log N/\log \tilde{d}$, where $\tilde{d} \coloneqq \sum_{i=1}^{N} w_i^2/\sum_{i=1}^{N} w_i$ is the second order average degree [112]. This matches the diameter of ER random graphs presented in section 3.1.1 when inserting $w_i = Np$. Due to the independent sampling of edges, random graphs with given expected degrees also lack a community structure and have a low clustering coefficient.

Although the expected degrees are specified in this model, the actual degree of each node varies in every realization. If the random graph should reproduce the degrees exactly, the *configuration model*, which will be discussed in section 3.1.6, can be used instead. Furthermore, random graphs with given expected degrees can themselves be considered as a special case of an even more general model: the *graphon model*. In the graphon model the probabilities $p_{i,j}$ for every edge $\{i,j\}$ are dictated instead of the expected degrees $w_i = \sum_{j=1}^{N} p_{i,j}$. Further details on the graphon model will be presented later in section 3.2.

3.1.4. The Watts–Strogatz small-world model

Motivated to construct networks that exhibit the *small-world property*, i.e., have a small diameter and a large clustering coefficient, Watts and Strogatz proposed the following model [106]. Given the number of nodes N and the mean degree K, which has to be an even number, a regular ring lattice is constructed such that every node is connected to its nearest $K/2$ left and $K/2$ right neighbors on the ring. Then, for every node i iterate through each of its $K/2$ right neighbors, and rewire it with probability β. Rewiring is done by replacing the edge $\{i,j\}$ with $\{i,k\}$, where k is chosen uniformly at random from all possible nodes, but avoiding self-loops and link duplication.

By varying the rewiring probability β it is possible to interpolate between a very regular lattice structure ($\beta = 0$), which exhibits a large clustering coefficient and large diameter, and a random structure ($\beta = 1$, close to an ER random graph), which exhibits a small clustering coefficient and small diameter. When increasing β, the diameter decreases much quicker than the clustering coefficient [106]. Hence, for intermediate choices of β the diameter is already small while the clustering coefficient is still comparatively large: the small-world property. However, similarly to ER random graphs, the degree distribution is concentrated, leading to networks where all nodes have similar degrees. Thus, the Watts–Strogatz model is not able to generate realistic networks with heavy-tailed degree distributions.

As mentioned before, the resulting random graph for $\beta = 1$ is similar to an ER random graph with $p = \frac{K}{N-1}$, but not identical. To generate exactly the same distribution, the generation algorithm can be slightly modified. The resulting random graph model will be called the *binomial Watts–Strogatz model*, see algorithm 1. In the binomial Watts–Strogatz model, the probability of an edge $\{i,j\}$ is $(1-\beta)+\beta\tilde{\beta}$ if i and j were connected originally on the ring lattice, and $\tilde{\beta}$ else. The value $\tilde{\beta}$ is chosen such that setting $\beta = 1$ results in an Erdős–Rényi model with $p = \frac{K}{N-1}$. Moreover, for every choice of β the expected degree of every node is K.

Algorithm 1 Binomial Watts–Strogatz model

```
Construct a regular ring lattice with $N$ nodes, such that each node is connected to its $K/2$ right and $K/2$ left neighbors.
Iterate through all $KN/2$ present edges and remove each with probability $\beta$. Denote the resulting graph as $G_1$.
Sample an ER random graph with edge probability
    $$\tilde{\beta} := \frac{\beta \frac{KN}{2}}{\binom{N}{2} - (1-\beta)\frac{KN}{2}} = \frac{\beta K}{N - 1 - (1-\beta)K}.$$
    Denote the resulting graph as $G_2$.
Return $G = G_1 \cup G_2$. ($G$ also has nodes $[N]$, and an edge is present in $G$ if it is present in $G_1$ or $G_2$.)
```

3.1.5. The Albert–Barabási model

The random graph models described in the previous sections all possess concentrated degree distribution that are not heavy-tailed. The *Albert–Barabási model* [113] on the other hand is specifically tailored to generate scale-free random networks with a power law degree distribution. Given a parameter $m \in \mathbb{N}$, the model starts with a small initial graph of at least m nodes, typically a clique or a star graph. Then nodes are iteratively added to the graph using the *preferential attachment* method until a desired number of N nodes is reached. In the preferential attachment method a new node is connected to m randomly chosen existing nodes. The probability p_i of picking the existing node i to form a new edge is proportional to its degree d_i, i.e.,

$$p_i = \frac{d_i}{\sum_j d_j}. \tag{3.5}$$

Hence, nodes with high degree are likely to accumulate even more edges as the graph grows, which results in the formation of the characteristic network hubs. Independently of the parameter m, this procedure results (asymptotically) in a power law degree distribution

$$\mathbb{P}(\boldsymbol{d} = k) \sim k^{-3}, \tag{3.6}$$

where $\boldsymbol{d}$ is the degree of a uniformly randomly chosen node [113].

The diameter of the Albert–Barabási model grows slowly at the rate $\frac{\log N}{\log \log N}$ for $m \geq 2$ and $\log N$ for $m = 1$ [114], and can thus be considered as small. However, the average clustering coefficient decreases quickly at the rate $\frac{\log^2 N}{N}$ [115], such that the model does not satisfy the small-world property. Moreover, due to the stochastic correlation of node degrees and edges, the Albert–Barabási random graph is substantially more difficult to analyze than simpler models like the Erdős–Rényi random graph.

3.1.6. The configuration model

Generating a random graph with an arbitrary degree distribution is possible with the *configuration model* [116]. Given the desired degree d_i for each node $i \in [N]$, such that $D := \sum_{i=1}^{N} d_i$ is even, d_i *half-edges* are assigned to each node i. Then repeatedly two half-edges are picked uniformly at random from all remaining half-edges, joined together to form an edge between the nodes they were assigned to, and removed from the set of available half-edges. This procedure may yield a *multigraph* that contains self-loops and multiple edges between two nodes, in which case the algorithm has to be rerun until a simple graph is sampled.

A more precise definition of the configuration model is as follows [109]. Define the set

$$W = [D] = W_1 \sqcup \cdots \sqcup W_N \tag{3.7}$$

with partition $W_1 = \{1, \ldots, d_1\}$, $W_2 = \{d_1 + 1, \ldots, d_1 + d_2\}$, and so on. The elements of W_i label the half-edges attached to node i, and W is an enumeration of all half-edges. Moreover, define the map

$$\varphi : W \to [N], \quad \varphi(e) = i \Leftrightarrow e \in W_i. \tag{3.8}$$

Joining two half-edges $e, h \in W$ corresponds to adding an edge between the nodes $\varphi(e)$ and $\varphi(h)$. A *configuration* is defined as a partition F of W into $\eta := D/2$ pairs. The multigraph that is induced by the configuration F is denoted by

$$\gamma(F) := (V, E), \quad V = [N], \quad E = \{\{\varphi(e), \varphi(h)\} \mid \{e, h\} \in F\}. \tag{3.9}$$

Let the random variable $\boldsymbol{F}$ denote a uniformly random configuration, i.e., every configuration is equally likely. Then the output of the configuration model is $\gamma(\boldsymbol{F})$.

When repeatedly generating multigraphs with the configuration model until a simple graph is sampled, each possible simple graph in $\mathcal{G}_N$ with the required degree distribution is equally likely to occur [109, Cor. 9.2]. Hence, the (simple) random graph induced by this procedure samples uniformly over the set of valid graphs. Moreover, under the sparsity condition $\max\{d_1, \ldots, d_N\} < N^{1/7}$, it can be shown that [109, Thm. 9.3]

$$\mathbb{P}(\gamma(\boldsymbol{F}) \text{ is simple}) \approx e^{-\lambda(\lambda+1)}, \quad \text{with } \lambda := \frac{\sum_{i=1}^N d_i(d_i - 1)}{2\sum_{i=1}^N d_i}. \tag{3.10}$$

Thus, in many cases only very few samples are sufficient to obtain a simple graph from the configuration model.

It remains to find a feasible way to sample configurations F uniformly at random. Define the set of η-tuples

$$\Pi := \Big\{(t_1, \ldots, t_\eta) \mid t_r \in [D - 2r + 1] \text{ for all } r\Big\}. \tag{3.11}$$

A tuple $t \in \Pi$ uniquely induces a configuration $F = \psi(t)$, where the map ψ is defined via the following algorithm. Let $U^1 := W$ and $U^{r+1} := U^r \setminus \{u_0^r, u_{t_r}^r\}$, $r = 1, \ldots, \eta$, where $u_i^r :=$ $(i+1)$-th smallest element of U^r. Then $F = \psi(t) := \{(u_0^r, u_{t_r}^r) \mid r = 1, \ldots, \eta\}$. In words, start with W and define an edge by connecting the nodes associated to the first and t_1-th element of W. Then remove this pair of elements from W and continue this procedure on the remaining set with the first and t_2-th element, and so on. An example is shown in Figure 3.1. Let the random variable $\boldsymbol{t}$ sample uniformly from Π, i.e., each component t_r is picked uniformly at random and independently of the others. Then it follows that

$$\psi(\boldsymbol{t}) \overset{d}{=} \boldsymbol{F}. \tag{3.12}$$

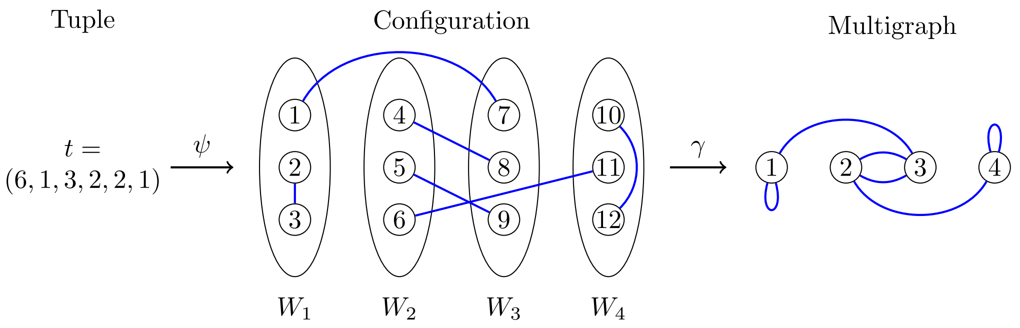


Figure 3.1.: For $N = 4$, $(d_1, \dots, d_4) = (3, \dots, 3)$, and an example tuple $t \in \Pi$, the induced configuration $\psi(t)$ and multigraph $\gamma(\psi(t))$ are shown. Reprinted from [1].

The averaged local clustering coefficient C of random graphs generated by the configuration model is given by [103]

$$C \approx \frac{\lambda}{N}, \quad \text{with } \lambda := \frac{\left(\frac{1}{N}\sum_{i=1}^{N} d_i^2 - \frac{1}{N}\sum_{i=1}^{N} d_i\right)^2}{\left(\frac{1}{N}\sum_{i=1}^{N} d_i\right)^3}. \tag{3.13}$$

Moreover, the random graph is a.a.s. connected if $d_i \geq 3$ for all i, although this condition is not necessary for connectedness [117]. If the degrees are additionally chosen such that the degree distribution converges to a limiting distribution with finite first and second moments, the diameter fulfills [118]

$$\operatorname{diameter}\bigl(\gamma(\boldsymbol{F})\bigr) = \alpha \log N + o(\log N) \quad \text{as } N \to \infty, \tag{3.14}$$

where α is some constant depending on the degree distribution. Hence, for these sparse random graphs generated by the configuration model, the diameter increases as $\log N$ while the clustering coefficient decreases as $1/N$, and they therefore do not fulfill the small-world property.

An interesting class of random graphs are the *uniformly random d-regular graphs*, where every d-regular graph has the same probability. A graph is called d-regular if every node i has degree $d_i = d$. As this constitutes a special case of the configuration model, the properties described above can be transferred to the random regular graphs. For instance, given $d \geq 3$, the resulting diameter is given by [119]

$$\operatorname{diameter}\bigl(\gamma(\boldsymbol{F})\bigr) = \frac{\log N}{\log(d-1)} + o(\log N) \quad \text{as } N \to \infty. \tag{3.15}$$

Due to the uniformly random drawing of (half-) edges in the configuration model, the resulting random graphs have a very homogeneous edge density when controlling for the dictated node degrees, which is also the reason for the small clustering coefficient discussed above. Thus, especially for large given degrees the random graphs generated by the configuration model are structurally quite similar to ER random graphs. For instance, the distributions of the random regular graph for a sufficiently large degree d and the ER random graph with $p = d/N$ become asymptotically identical, which is the subject of the *sandwich conjecture* [120]:

Conjecture 3.4. *If the degree $d = d(N)$ is a function of the graph size N that satisfies $d \gg \log N$, then there exist $p_* = (1 - o(1))\, d/N$ and $p^* = (1 + o(1))\, d/N$ as well as ER random graphs $\boldsymbol{G}_* \sim \boldsymbol{G}_{N,p_*}$ and $\boldsymbol{G}^* \sim \boldsymbol{G}_{N,p^*}$, such that*

$$\mathbb{P}(\boldsymbol{G}_* \subseteq \boldsymbol{G}_{N,d} \subseteq \boldsymbol{G}^*) = 1 - o(1) \quad \text{as } N \to \infty, \tag{3.16}$$

where $\subseteq$ denotes inclusion of edges.

Proof. The sandwich conjecture has only been proven for $d \gg \log(N)^4 / \log(\log N)^3$, see [121]. It is an open question whether or not the conjecture is true for the missing range $\log N \ll d \ll \log^4 N / \log^3 \log N$. □

3.2. Graphons

A useful tool for working with random graphs are *graphons* [122] (short for graph-functions), which are commonly employed in the study of large and dense networks.

Definition 3.5. A measurable and symmetric function $W : [0,1]^2 \to [0,1]$ is called a *graphon*.

For all $N \in \mathbb{N}$ the graphon W induces a random graph $\boldsymbol{G}_N$ on N nodes by the following procedure. Define $u_i := i/N$ for $i = 1, \dots, N$. Then the edge $\{i, j\}$ is present in $\boldsymbol{G}_N$ with probability $W(u_i, u_j)$, independently of other edges. Note that the expected number of edges $|\boldsymbol{E}_N|$ in $\boldsymbol{G}_N$ is asymptotically

$$\mathbb{E}[|\boldsymbol{E}_N|] \approx \frac{N^2}{2} \int_0^1 \int_0^1 W(x, y) \mathrm{d}x \, \mathrm{d}y. \tag{3.17}$$

Thus, the random graphs induced by a graphon are dense.

Furthermore, given a graph G an associated graphon W^G can be specified. Define the interval $I_i := (u_{i-1}, u_i]$, and set $W^G(I_i \times I_j) \equiv 1$ if the edge $\{i, j\}$ is presented in G, and $W^G(I_i \times I_j) \equiv 0$ else. Due to this block structure, the graphon W^G can be seen as a functional representation of the adjacency matrix of G.

Many popular random graph models can be represented as a graphon. The simplest example is $W \equiv p$, which induces the Erdős–Rényi random graph $\boldsymbol{G}_{N,p}$ (see section 3.1.1). Moreover, partitioning $[0,1]$ into intervals $J_1, \dots, J_K$ and setting $W(J_k \times J_{k^*}) \equiv P_{k,k^*}$ yields the stochastic block model (see section 3.1.2) with K communities, whose sizes are proportional to the length of the associated intervals, and probability matrix P. However, due to the fact that edges are always independent in the induced random graph, any model with stochastically dependent edges cannot be represented as a graphon. For example, the Albert–Barabási model (section 3.1.5) and the configuration model (section 3.1.6) lack a graphon representation.

Graphon operator. Given a graphon W, the *graphon operator* $\mathbb{W}$, which acts on functions $f \in L^1([0,1])$, is defined by

$$(\mathbb{W} f)(x) = \int_0^1 W(x, y) f(y) \mathrm{d}y. \tag{3.18}$$

For the important special case $f \equiv 1$ the graphon operator yields the degree function

$$d_W(x) \coloneqq (\mathbb{W}1)(x) = \int_0^1 W(x,y)\mathrm{d}y. \tag{3.19}$$

It describes how many neighbors a node at position x has on average asymptotically, from $d_W(x) = 1$ meaning that it is connected to all other nodes to $d_W(x) = 0$ implying that it is isolated. For large finite networks the expected degree of node i is given by

$$\mathbb{E}[\boldsymbol{d}_i] = \sum_{j \neq i} W(u_i, u_j) \approx N\, d_W(u_i). \tag{3.20}$$

Graph limits. Graphons can also be used to define the limit of a sequence of (random) graphs. The distance between graphs is typically measured in the *cut norm*, although other sensible choices exist [122]. The cut norm of a graphon W is defined as

$$\|W\|_\square \coloneqq \sup_{S,T \subset [0,1]} \left| \int_S \int_T W(x,y)\mathrm{d}x\,\mathrm{d}y \right|. \tag{3.21}$$

Hence, given two graphs G and H, and their associated graphons W^G and W^H, the distance $\|W^G - W^H\|_\square$ measures the maximum discrepancy of the edge density between G and H over all subsets of nodes.

Let G_ℓ be a graph on N_ℓ nodes for all $\ell \in \mathbb{N}$ and assume that $N_\ell \to \infty$. A graphon W is called the *graph limit* of the sequence of graphs $(G_\ell)_\ell$ if $\lim_{\ell\to\infty}\|W^{G_\ell} - W\|_\square = 0$. For example, let G_ℓ be a realization of the Erdős–Rényi random graph $\boldsymbol{G}_{N_\ell,p}$, and $W \equiv p$. Then it can be shown [122] that almost surely W is the graph limit of $(G_\ell)_\ell$.

To circumvent the density restriction of graphons and define the graph limit of a sequence of non-dense graphs, a scaling parameter $\kappa_\ell > 0$ can be used [85]. More precisely, the convergence criterion is modified to $\lim_{\ell\to\infty}\|\frac{1}{\kappa_\ell}W^{G_\ell} - W\|_\square = 0$. For instance, if G_ℓ is a realization of the ER random graph $\boldsymbol{G}_{N_\ell,\kappa_\ell}$ with $\kappa_\ell \to 0$, the sequence $(G_\ell)_\ell$ has almost surely the rescaled graph limit $W \equiv 1$.

Applications and Extensions. Graphons have been applied extensively in the study of dynamical systems on networks (or interacting particle systems) [85, 123, 124], but also in adjacent fields like control theory on networks [125] or dynamical game systems on networks [126]. Graphons are especially well-suited for deriving a mean-field limit in the form of a PDE, for instance a McKean–Vlasov or continuity equation, that describes the evolution of probability densities. The above-mentioned graphon operator $\mathbb{W}$ often plays a central role in these mean-field equations. An example application of this graphon mean-field theory to the voter model will be demonstrated in section 7.3.

As graph limits of *sparse* networks cannot be handled using the notion of graphons as introduced above, different theories have been developed to define sparse graph limits, see for example [127, 128, 129]. A promising approach to unify the theories for dense and sparse graph limits is to assess and compare graphs based only on the *action* of the operator given by their adjacency matrix [130]. These positivity-preserving and self-adjoint operators are called *graphops* (short for graph-operators). Under mild conditions on a sequence of graphs, the associated sequence of graphops converges to a limit operator in a certain metric that is related to their action and utilizes a concept called *profiles*.

For instance, in the case that a sequence of dense graphs converges to a graphon, the associated limit graphop is the graphon operator $\mathbb{W}$. Mean-field limits for dynamical systems on networks have also recently been formulated using graphops, see [131, 132].

3.3. Invariance under graph isomorphism

A useful property of some random graphs, which will be important later in section 5.2, is the *invariance under graph isomorphism* [1]. Let $\mathcal{G}_N$ denote the set of simple graphs with node set $[N]$. A graph isomorphism between two simple graphs $G = ([N], E_G)$ and $H = ([N], E_H)$ is a permutation $\tau : [N] \to [N]$ such that

$$(i, j) \in E_G \quad \Leftrightarrow \quad (\tau(i), \tau(j)) \in E_H. \tag{3.22}$$

Hence, we will denote $H = \tau(G)$ if H and G are isomorphic with permutation τ. Many common random graph models are indifferent with respect to the specific node labels, which motivates the following definition:

Definition 3.6. A random graph $\boldsymbol{G} \in \mathcal{G}_N$ is called *invariant under isomorphism* if for any two isomorphic graphs $G, H \in \mathcal{G}_N$ it holds

$$\mathbb{P}(\boldsymbol{G} = G) = \mathbb{P}(\boldsymbol{G} = H). \tag{3.23}$$

Example 3.7. Erdős–Rényi random graphs (see section 3.1.1) are invariant under isomorphism as the probability of a graph depends only on the number of its edges, which is preserved under graph isomorphism. The uniformly random d-regular graph (section 3.1.6) is also invariant under isomorphism as every d-regular graph has equal probability, any not d-regular graph has probability 0, and d-regularity is preserved under graph isomorphism.

It is interesting to examine this invariance of the random graph together with a system state $x \in [M]^N$, which assigns to every node i a discrete state $x_i \in [M]$. For a permutation $\tau : [N] \to [N]$, the permuted state $\tau(x) \in [M]^N$ is defined by $\tau(x)_i := x_{\tau^{-1}(i)}$. Then, certain observables f are identical for (G, x) and $(\tau(G), \tau(x))$, for example the number of edges between nodes of state m and nodes of state n.

Definition 3.8. A function $f : \mathcal{G}_N \times [M]^N \to \mathbb{R}$ is called *invariant under isomorphism* if for all permutations $\tau : [N] \to [N]$ and all $(G, x) \in \mathcal{G}_N \times [M]^N$ it holds $f(G, x) = f(\tau(G), \tau(x))$.

Example 3.9. Let $f(G, x)$ denote the number of edges between nodes of state m and nodes of state n, $m \neq n$, and $d^G_{i,n}(x)$ the number of neighbors of node i that are of state n. Then it follows

$$\begin{aligned} f(G, x) &= \sum_{i\colon x_i = m} d^G_{i,n}(x) && (3.24)\\ &= \sum_{i\colon x_i = m} d^{\tau(G)}_{\tau(i),n}(\tau(x)) && (3.25)\\ &= \sum_{i\colon \tau(x)_i = m} d^{\tau(G)}_{i,n}(\tau(x)) = f(\tau(G), \tau(x)), && (3.26) \end{aligned}$$

i.e., $f(G, x)$ is invariant under isomorphism.

Consider the following useful proposition.

Proposition 3.10. *Let both the random graph $\boldsymbol{G} \in \mathcal{G}_N$ and the function $f : \mathcal{G}_N \times [M]^N \to \mathbb{R}$ be invariant under isomorphism. Let $\tau : [N] \to [N]$ be a permutation and $x \in [M]^N$ a state. Then it holds*

$$f(\boldsymbol{G}, x) \overset{d}{=} f(\boldsymbol{G}, \tau(x)). \tag{3.27}$$

Proof. Define for some fixed $\beta \in \mathbb{R}$

$$\mathbb{G} \coloneqq \{G \in \mathcal{G}_N \ : \ f(G, x) = \beta\} \tag{3.28}$$

$$\mathbb{G}^* \coloneqq \{G \in \mathcal{G}_N \ : \ f(G, \tau(x)) = \beta\}. \tag{3.29}$$

Due to the invariance under isomorphism of f, any $G \in \mathbb{G}$ satisfies

$$\beta = f(G, x) = f(\tau(G), \tau(x)), \tag{3.30}$$

and thus $\tau(G) \in \mathbb{G}^*$. Now, let $G^* \in \mathbb{G}^*$. Then

$$f(\tau^{-1}(G^*), x) = f(G^*, \tau(x)) = \beta, \tag{3.31}$$

and thus $\tau^{-1}(G^*) \in \mathbb{G}$. Altogether, we have $\tau(\mathbb{G}) = \mathbb{G}^*$. Finally, by the invariance under isomorphism of $\boldsymbol{G}$, it follows

$$\mathbb{P}(f(\boldsymbol{G}, x) = \beta) = \mathbb{P}(\boldsymbol{G} \in \mathbb{G}) = \mathbb{P}(\boldsymbol{G} \in \mathbb{G}^*) = \mathbb{P}(f(\boldsymbol{G}, \tau(x)) = \beta). \tag{3.32}$$

□

4. Simulation of Markov jump processes on networks

Many popular discrete-state dynamical models on networks fall into the class of Markov jump processes, including the continuous-time noisy voter model that was introduced in section 2.2. Due to the complexity of these systems, a purely theoretical investigation of their behavior is often out of reach. Instead, numerical simulations are readily employed to generate system trajectories, to conduct statistical tests of proposed hypotheses about the system's characteristics, or to assess the quality of model reductions. Section 4.1 explores how these Markov jump processes on networks can be simulated efficiently on a computer. Moreover, a specific simulation algorithm for the continuous-time noisy voter model is developed in section 4.2.

Consider again a simple graph G with N nodes. Each node $i \in [N]$ has one of $M \in \mathbb{N}$ discrete states, $x_i \in [M]$. It is assumed that the underlying model is such that each node i changes its state over time due to a (inhomogeneous) continuous-time Markov chain with transition rate matrix $Q_i(x)$, $Q_i : [M]^N \to \mathbb{R}^{M \times M}$. For $m \neq n$, the (m, n)-th entry of the rate matrix, $(Q_i(x))_{m,n} \geq 0$, specifies at which rate node i transitions from state m to state n. The diagonal entries are such that each row sums to 0. Note that the transition rates $Q_i(x)$ may depend on the full system state x, although in most popular models they are calculated based only on the neighborhood of agent i. Moreover, each node i may be subject to a different function Q_i determining its transition rates. The stochastic process referring to the state of node i at time t is denoted as $\boldsymbol{x}_i(t)$, and the full process as $\boldsymbol{x}(t) = \big(\boldsymbol{x}_1(t), \dots, \boldsymbol{x}_N(t)\big) \in [M]^N$. The full system is then a time-homogeneous Markov jump process with generator Q, which specifies the rate of transitioning to a state $y \in [M]^N$ when starting in a different state $x \in [M]^N$ and is given by

$$Q(x, y) = \begin{cases} \big(Q_i(x)\big)_{x_i, y_i} & \exists i \in [N] \ \forall j \neq i : x_j = y_j \\ 0 & \text{else.} \end{cases} \tag{4.1}$$

This process has the following interpretation: Given $\boldsymbol{x}(t) = x$, the processes $\boldsymbol{x}_i(s)$, $i = 1, \dots, N$, are independent continuous-time Markov chains with rate matrices $Q_i(x)$ for $s > t$ as long as no jump takes place, i.e., $\boldsymbol{x}(r) = x$ for all $t < r < s$. When a jump occurs (almost surely no two jumps occur simultaneously), the state x is updated, with it the rate matrices $Q_i(x)$, and the independent processes in the nodes start over with initial conditions dictated by the updated state x.

4.1. The stochastic simulation algorithm

At first, the numerical simulation of a general continuous-time Markov chain is discussed. This is extended to the special case of Markov jump processes on networks later in the section.

An obvious but inefficient way to sample realizations of a general Markov jump process with rate matrix Q is given by the following procedure. Given an initial state m of the process at time t, draw for each other state $n \neq m$ a random time $\Delta_n \sim \exp(Q_{m,n})$ that is exponentially distributed with the corresponding transition rate.[1] Then at time $t + \Delta$, $\Delta := \min_k \Delta_k$, the process jumps to the new state $n := \text{argmin}_k \Delta_k$, and the procedure is started anew. As a time Δ_k has to be sampled for each n and the minimum needs to be calculated, this approach suffers from a linear complexity in the number of states.

A much more efficient procedure is given by the *stochastic simulation algorithm* (SSA; also called Gillespie's algorithm) [134], which exploits the property that the minimum of independent exponentially distributed variables is again exponentially distributed, with accumulated rate. In the SSA, the duration until the next transition event is drawn from an exponential distribution corresponding to the accumulated rate, and then the new state is drawn randomly with probabilities proportional to the associated transition rates, see algorithm 2. As each row of the rate matrix sums to 0, these accumulated rates are given by the diagonal entries of Q.

Algorithm 2 Stochastic simulation algorithm (SSA)

```
Input: rate matrix $Q$, state $m$, time $t$
1: draw $\Delta \sim \exp(-Q_{m,m})$
2: $t \leftarrow t + \Delta$
3: draw $n$ according to $\mathbb{P}(n = k) = \frac{Q_{m,k}}{-Q_{m,m}}$
4: $m \leftarrow n$
```

An important step of the SSA is the sampling of the new state n according to the magnitude of the respective transition rates (line 3 in algorithm 2). There are several approaches one can employ to sample from such a non-uniform discrete distribution. A popular method is sampling a uniformly distributed random number $u \in [0, 1]$ and then finding the smallest n such that $F(n) \geq u$, where F denotes the cumulative distribution function

$$F(n) = \sum_{k=1}^{n} \frac{Q_{m,k}}{-Q_{m,m}}. \tag{4.2}$$

This produces samples n with the correct distribution [133], and the values $F(1), F(2), \ldots$ can be calculated beforehand. Determining the smallest n with $F(n) \geq u$ can be achieved via a bisection approach, which results in a logarithmic complexity with respect to the number of states. However, by utilizing a so-called *index table*, which requires some additional setup cost, the correct n can be found with approximately constant complexity [135].

A different family of methods for sampling from discrete distributions is based on *lookup tables*. For example, if there are two states $\{1, 2\}$ such that $\mathbb{P}(n = 1) = \frac{1}{4}$ and $\mathbb{P}(n = 2) = \frac{3}{4}$, a uniform sample from the array (table) $[1, 2, 2, 2]$ would generate the desired distribution. Constructing these tables for a large number of states with potentially inconvenient (e.g, irrational) probabilities poses many difficulties, but there are algorithms that are able to utilize lookup tables for arbitrary distributions [136], again resulting in a fast constant complexity sampling after some additional setup cost.

[1] Exponentially distributed samples can for example be generated from a uniform distribution via inverse transformation [133].

The algorithm that will be employed here is the *alias method* [137], which requires two tables: the probability table U containing a probability $U_n \in [0, 1]$ for each state n, and the alias table K containing an alias state K_n for each state n. A sample is generated by uniformly drawing a state n and a number $u \in [0, 1]$. Then, if $u < U_n$, the state n is returned. Otherwise, the alias state K_n is returned. For a given distribution, there can be multiple correct choices for the probability table and alias table. The optimal choice is characterized by the probability table with the largest sum of entries as this minimizes the number of lookups of the alias table. Generating (non-optimal) tables can be achieved by the following simple procedure. First initialize the probability table with values $U_n = S\,p_n$, where S is the total number of states and $p_n = -Q_{m,n}/Q_{m,m}$ the desired probability of state n. Then pick a state n_1 with $U_{n_1} > 1$ and a state n_2 with $U_{n_2} < 1$. Shift the excess probability mass of n_1 to n_2 by setting n_1 as the alias of n_2, i.e., $K_{n_2} = n_1$, and decrease the probability U_{n_1} to $U_{n_1} - (1 - U_{n_2})$ to maintain the correct distribution. Repeat this procedure until all states have $U_n \leq 1$.

By applying a sampling algorithm with constant time complexity like the alias method, the complete SSA update step (cf. algorithm 2) also exhibits a constant complexity with respect to the number of states S. However, some additional work is necessary once at the start to set up the sampling method, typically with linear complexity in S per initial state m. For instance, in the alias method the construction of the two tables using the algorithm described above is $\mathcal{O}(S)$, and it has to be done for each initial state m, leading to a total complexity of $\mathcal{O}(S^2)$. Note also that the resulting memory requirement of $\mathcal{O}(S^2)$ is identical to that of the bisection approach presented in (4.2) if all values $F(n)$ of the cumulative distribution function are precomputed for each associated initial state.

Applying the SSA to Markov jump processes on networks. For the Markov processes on networks described in the beginning of the chapter, the plain SSA is computationally not feasible for all but very small networks. The generator Q (4.1) can be interpreted as a rate matrix of dimension $M^N \times M^N$ by enumerating the states $x \in [M]^N$. This rate matrix is quite sparse as only transition rates between two states that differ in exactly one coordinate can be nonzero. Hence, in each row of Q the number of nonzero elements is at most $N(M-1)$. Nevertheless, computing or storing all the transition rates beforehand is not feasible due to the huge number M^N of different states. (For instance, if $M = 2$ and the graph has $N = 100$ nodes, then the number of nonzero entries is $M^N N(M-1) \approx 10^{32}$.) As a consequence, also the calculation of lookup tables for constant complexity sampling is not feasible.

This issue can be circumvented by only computing the necessary rates during the SSA update step, i.e., if the system is in a given state, compute the transition rates to the $N(M-1)$ possible next states, calculate their sum to obtain the accumulated rate, and execute the SSA update from algorithm 2. Unfortunately, the repeating calculation of the accumulated rate in the update step again implies a linear time complexity in the number of nodes N. As the expected number of update steps required to simulate a given time horizon also increases linearly with N (assuming that each node has identical rates Q_i), the complexity of the whole simulation scales with N^2M, which again makes this approach unfeasible for larger networks.

A possible solution is to employ an approximation technique such as τ-*leaping* [41] that offers better performance at the cost of accuracy. Instead of simulating every single transition like in the SSA, the τ-leaping method aims to aggregate all jumps that occur in a short time τ into one update step. This introduces an error whose magnitude depends on the size of τ, but it removes the scaling of the number of required update steps in a given time horizon with N.

However, given a specific model with known transition rates there are a number of tricks that can be employed to potentially improve the performance of the SSA significantly. Using the SSA compared to approximation methods like τ-leaping should be preferred if it is computationally feasible because the SSA produces statistically correct samples of the process. The next section illustrates some modifications to the SSA for the continuous-time noisy voter model, making simulations with more than a hundred thousand agents viable.

4.2. Simulating the continuous-time noisy voter model

In the continuous-time noisy voter model (CNVM), which was discussed in detail in section 2.2, each node i switches from one opinion $m \in [M]$ to a different opinion $n \in [M]$ at the rate

$$(Q_i(x))_{m,n} := r_{m,n} \frac{d_{i,n}(x)}{(d_i)^\alpha} + \tilde{r}_{m,n}, \tag{4.3}$$

where $d_{i,n}(x)$ denotes the number of neighbors of node i with opinion n, d_i is the degree of node i, and $r_{m,n}, \tilde{r}_{m,n} \geq 0$ and $\alpha \in \mathbb{R}$ are model parameters. For an easier formulation of the stochastic simulation algorithm (SSA), the noise parameters $\tilde{r}_{m,n}$ are at first assumed to be 0, but the general case will be discussed later.

It is advantageous to rewrite the rates as follows. The parameter $r_{m,n}$ is replaced by $\hat{r}\, p_{m,n}$, where $\hat{r} := \max_{m,n} r_{m,n} \geq 0$ and $p_{m,n} := r_{m,n}/\hat{r} \in [0,1]$. Moreover, $d_{i,n}(x)/d_i^\alpha$ is replaced by $d_i^{(1-\alpha)}\, d_{i,n}(x)/d_i$, which yields

$$(Q_i(x))_{m,n} = \hat{r} d_i^{(1-\alpha)} p_{m,n} \frac{d_{i,n}(x)}{d_i}. \tag{4.4}$$

Now the rates consist of a factor $\hat{r} d_i^{(1-\alpha)}$ that only depends on i, and two factors $p_{m,n} \in [0,1]$ and $d_{i,n}(x)/d_i \in [0,1]$ that can be interpreted as probabilities (see section 2.2). Thus, the accumulated rate can be easily bounded,

$$\sum_{i,m,n} (Q_i(x))_{m,n} \leq \sum_i \hat{r} d_i^{(1-\alpha)} =: r_{\text{acc}}, \tag{4.5}$$

and the bound r_{acc} is fixed and can be precomputed. This gives rise to the following modification of the SSA. The time of the next transition is sampled according to the rate r_{acc} and the agent i that undergoes the transition is drawn randomly with probability $d_i^{(1-\alpha)} / \sum_j d_j^{(1-\alpha)}$. Then each transition from agent i's current opinion m to a new opinion n has a probability of $p_{m,n}\, d_{i,n}(x)/d_i$ to occur. Note that the sum of these probabilities over n can be less than 1. The remaining probability mass corresponds to the case that no transition takes place. Hence, either a new opinion n is drawn for agent i or its opinion remains unchanged after the update step.

Although the costly calculation of the accumulated rate in each update step is averted by this modification due to the precalculation of r_{acc}, the evaluation of $d_{i,n}(x)/d_i$ for each n still constitutes a substantial cost (scaling with the degree of nodes). Luckily, another trick can be applied to circumvent this evaluation: the new opinion n can also be sampled by picking a random neighbor j of node i and setting n to its opinion x_j. Thus, the probability to pick each opinion n is $d_{i,n}(x)/d_i$, and sampling n exhibits a constant complexity because the neighborhoods of each node can be calculated beforehand. After sampling the new opinion n, the transition is conducted if and only if a coin toss with probability $p_{m,n}$ succeeds. This modified version of the SSA for the CNVM is summarized in algorithm 3. (Note that in this procedure also a neighbor j with the same state as agent i can be drawn, in which case the opinion of agent i remains unchanged irrespective of the subsequent coin toss.)

Algorithm 3 Stochastic simulation algorithm for the CNVM ($\tilde{r} = 0$)

```
1: draw $\Delta \sim \exp(r_{\text{acc}})$
2: $t \leftarrow t + \Delta$
3: draw $i$ from $\{1, \dots, N\}$ according to $\mathbb{P}(i = k) = \frac{\hat{r} d_k^{(1-\alpha)}}{r_{\text{acc}}}$
4: draw random neighbor $j$ of node $i$ uniformly
5: draw $u \sim \text{Unif}[0, 1]$
6: if $u < p_{x_i, x_j}$ then
7:     $x_i \leftarrow x_j$
8: end if
```

The modified SSA in algorithm 3 again has a constant complexity in the number of nodes, making it viable for large networks. (For the non-uniform sampling in line 3, the before mentioned *alias method* can for example be employed to achieve constant complexity.) It should be noted however that this tremendous decrease in complexity of the update step comes at a (small) price: due to the coin toss in line 6 of algorithm 3, the update step has a chance to not result in any state change, hence "wasting" an iteration. The number of wasted iterations is large if many $p_{m,n}$ are small, which occurs when the imitation rates $r_{m,n}$ differ substantially for different m, n. If on the other hand all $r_{m,n}$ are equal, no iterations are wasted as all $p_{m,n} = 1$.

The exploration rates $\tilde{r}_{m,n}$, which were assumed to be zero up to now, can easily be included in the above algorithm, retaining a constant complexity update. Rewriting the rates in an analogous way yields

$$\tilde{r}_{m,n} = \hat{r}_{\text{expl}}\ \tilde{p}_{m,n} \frac{1}{M}, \tag{4.6}$$

where $\hat{r}_{\text{expl}} := M \max_{m,n} \tilde{r}_{m,n}$ and $\tilde{p}_{m,n} := \tilde{r}_{m,n} / \max_{m,n} \tilde{r}_{m,n}$. Taking account of these additional rates in the calculation of the accumulated rate, a coin toss can be utilized to determine if the next transition is due to the influence of neighbors or due to exploration (noise). Algorithm 4 shows the complete simulation algorithm for the CNVM.

Performance. As mentioned before, the average number of state updates in a given time horizon increases linearly[2] with the number of nodes N. The expected number of iterations of algorithm 4 also increases linearly due to the linear growth of the accumulated rate r_{acc}. Because of the above-mentioned wasted iterations that do not yield a state

[2] Assuming that the newly added nodes have similar rates $r, \tilde{r}$ and, if $\alpha \neq 1$, also similar degrees d_i.

Algorithm 4 Simulation algorithm for the CNVM

```
Input: initial state $x$, end time $T$
$t \leftarrow 0$
$r_{\text{acc}} \leftarrow \sum_{i=1}^{N} \hat{r} d_i^{(1-\alpha)} + N\hat{r}_{\text{expl}}$        ▷ accumulated transition rate
$p_{\text{expl}} \leftarrow N\hat{r}_{\text{expl}}/r_{\text{acc}}$        ▷ probability for transition due to exploration
while $t < T$ do
    draw $\Delta \sim \exp(r_{\text{acc}})$
    $t \leftarrow t + \Delta$
    draw $u \sim \text{Unif}[0,1]$
    if $u < p_{\text{expl}}$ then        ▷ transition due to exploration
        draw $i$ from $\{1, \dots, N\}$ uniformly
        draw $n$ from $\{1, \dots, M\}$ uniformly
        draw $u \sim \text{Unif}[0,1]$
        if $u < \tilde{p}_{x_i,n}$ then
            $x_i \leftarrow n$
        end if
    else        ▷ transition due to imitation
        draw $i$ from $\{1, \dots, N\}$ according to $\mathbb{P}(i = k) = \frac{\hat{r} d_k^{(1-\alpha)}}{r_{\text{acc}}}$
        draw random neighbor $j$ of node $i$ uniformly
        draw $u \sim \text{Unif}[0,1]$
        if $u < p_{x_i,x_j}$ then
            $x_i \leftarrow x_j$
        end if
    end if
end while
```

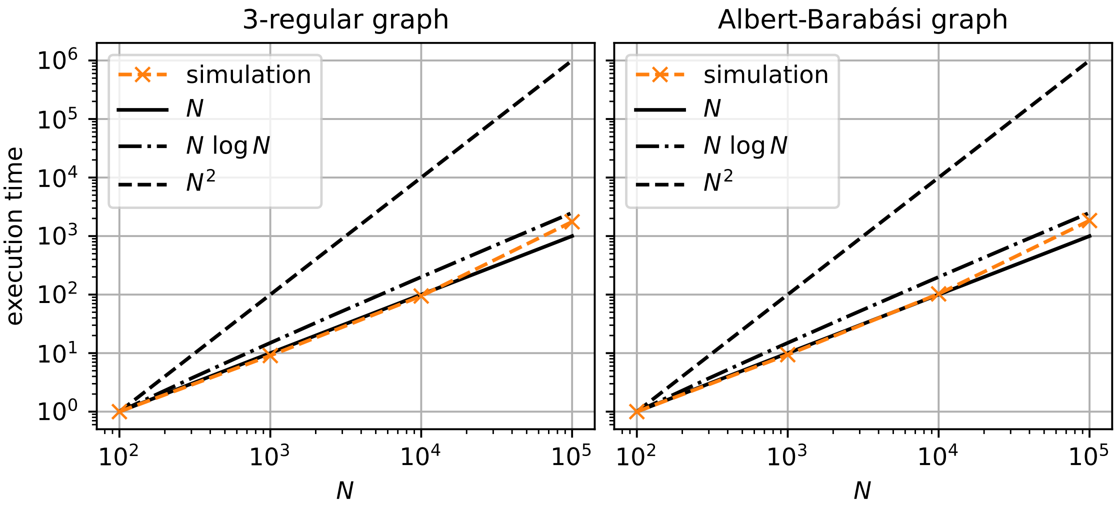


Figure 4.1.: Normalized execution time of the CNVM simulation (algorithm 4) for different numbers of nodes N and CNVM parameter $\alpha = 1$.

change, the expected number of iterations is an upper bound of the expected number of state updates, i.e., it may grow linearly with a larger than necessary factor. Anyhow, the overall expected execution time of algorithm 4 is still $\mathcal{O}(N)$ due to the constant complexity update step, and the larger than necessary factor due to the wasted iterations is negligible compared to the $\mathcal{O}(N^2)$ scaling of the standard SSA. The linear complexity of algorithm 4 is verified by numerical experiments conducted using the *SPoNet*[3] (Spreading Processes on Networks) Python package, which was developed by the author and is used for all simulations presented in this thesis. The code employs algorithm 4 together with the *alias method* for non-uniform sampling. Figure 4.1 illustrates the execution time for networks of different sizes, given a fixed end time T. Results for two different types of networks are shown: random 3-regular graphs (see section 3.1.6) and Albert–Barabási graphs (see section 3.1.5). As expected, the execution time increases linearly with N. However, for the largest examined networks ($N = 10^5$) performance seems to degrade, which is probably caused by the larger memory requirements, e.g., due to the size of the neighborhood data exceeding the size of the CPU cache.

[3] `https://github.com/lueckem/SPoNet`

5. Mean-field limits of Markov jump processes on random graphs

This chapter addresses the question whether certain Markov jump processes on networks possess a mean-field limit in the sense that the stochastic dynamics of the shares of each discrete state concentrate around an ordinary differential equation (ODE) in the large population limit.

The Markov jump processes discussed here are of the same type as in chapter 4: each node $i \in [N]$ of the simple graph G has one of $M \in \mathbb{N}$ discrete states, $x_i \in [M]$. It changes its state over time according to a (inhomogeneous) continuous-time Markov chain with transition rate matrix $Q_i(x)$, $Q_i : [M]^N \to \mathbb{R}^{M \times M}$. Again, the transition rates $Q_i(x)$ of node i may depend on the full system state x (although in most models they are calculated based only on the neighborhood of node i), and each node i may be subject to a different function Q_i. The stochastic process referring to the state of node i is denoted as $\boldsymbol{x}_i(t)$, and the full process as $\boldsymbol{x}(t) = \big(\boldsymbol{x}_1(t), \dots, \boldsymbol{x}_N(t)\big) \in [M]^N$. While the Markov chains describing the dynamics of single nodes are inhomogeneous, the global process $\boldsymbol{x}(t)$ is given by a homogeneous Markov chain with generator

$$Q(x, y) = \begin{cases} \big(Q_i(x)\big)_{x_i, y_i} & \exists i \in [N] \; \forall j \neq i : x_j = y_j \\ 0 & \text{else,} \end{cases} \tag{5.1}$$

describing the rate at which the global process transitions from a system state $x \in [M]^N$ to a different state $y \in [M]^N$. Many popular continuous-time discrete-state systems on networks can be defined within this framework, for instance voter models, including infection spreading models like the SIS or SIR model, and threshold models (see sections 2.1.2 and 2.2 for more details). In the following, a superscript will be employed to specify to which graph G a rate matrix is associated to, e.g. $Q_i^G(x)$.

Random graphs. The existence of a mean-field limit largely depends on the properties of the underlying networks. In this chapter, the above defined Markov jump processes are considered on *random graphs*. A random graph $\boldsymbol{G}$ with N nodes is defined as a random variable with values in the set of all $2^{N(N-1)/2}$ possible simple graphs with node set $[N]$, see chapter 3. The stochastic process $\boldsymbol{x}(t)$ then depends on both the random selection of an underlying graph according to $\boldsymbol{G}$ and the stochastic transitions of the Markov jump process. More precisely, a realization of $\boldsymbol{x}(t)$ is given by first sampling a graph G from $\boldsymbol{G}$, initializing the node states according to $\boldsymbol{x}(0)$, and then letting the Markov jump process run on G.

Random graphs are an important tool used to generate networks with certain characteristics, especially when real-world network data is difficult to obtain, making it interesting to consider them in combination with stochastic dynamics as defined above. Moreover, for the discussion of mean-field limits it will be necessary to define infinite increasing sequences of graphs, which is more easily done by using graph construction mechanisms like random graph models than by specifying each single graph in the sequence. Still, deterministic sequences of graphs are included as a special case in this setting and the results shown below can be applied to those sequences as well.

Collective variables and classes. As explained in section 2.4.1, the most important collective variables in the context of mean-field theory are the *shares* or *concentrations* of the different states, for instance the percentage of infected nodes in an epidemiological model. Instead of only considering the global shares in the whole system, it will be allowed to measure the shares separately for certain subsets of nodes, leading to a more versatile set of collective variables. In order to define these subsets, each node is assigned to one of $K \in \mathbb{N}$ classes. The classification of node i is denoted as $s_i \in [K]$, and is fixed over time and does not depend on the realization of the random graph[1]. The tuple (x_i, s_i) is called the *extended state* of node i. Finally, the collective variable $C : [M]^N \to \mathbb{R}^{MK}$ is defined by measuring the shares of each extended state, i.e.,

$$C(x) = \Big(C_{(m,k)}(x)\Big)_{m\in[M],k\in[K]}, \quad C_{(m,k)}(x) := \frac{\#\{i \in [N] : (x_i, s_i) = (m, k)\}}{N}. \tag{5.2}$$

Mean-field approximations, i.e., expressing the dynamics only in terms of these concentrations, work best if nodes are as indistinguishable and interchangeable as possible, see section 2.4.1 for details. Hence, these classes should be utilized to group nodes together if they have similar traits or similar functions in the network, and thus may become indistinguishable in the large population limit. Some examples of the choice of classes are presented below.

Example 5.1. Common examples for the choice of classes:

1. In the case $K = 1$, the extended states are the states themselves, $(x_i, s_i) = (x_i, 1) \cong x_i$. The collective variable $C(x)$ measures the global share of each state in the system, which is the most commonly discussed setting in mean-field literature, see section 2.4.1. A necessary and sufficient condition on the random graph sequence such that a mean field limit exists has been derived in [138] for processes where the transition rates Q_i are affine-linear in the so-called neighborhood vector. However, the continuous-time noisy voter model does not fall into this category of processes and neither do so-called "complex contagion" models in which the infection rates are nonlinear functions of the infection prevalence in the node's neighborhood.

2. If the random graph exhibits a fixed modular structure with K communities or clusters, it is a natural choice to measure the shares of the states in each cluster separately, i.e., a node located in cluster k is assigned to class k. Hence, the extended state $(x_i, s_i) = (m, k)$ refers to a node with state $x_i = m$ located in cluster k. An example is shown in figure 5.1. This technique of differentiating nodes by their community is frequently used in the literature, see for example [139, 140, 54]. Random graphs exhibiting this clustered structure can for instance be generated using the stochastic block model, see section 3.1.2.

[1] The restriction that classes must not depend on the realization of the random graph is made for ease of presentation. It can be lifted at the cost of more technical notation and proofs.

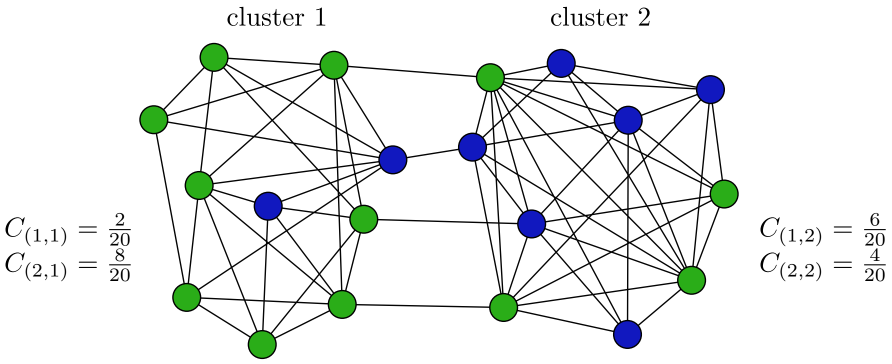

cluster 1
cluster 2
$C_{(1,1)} = \frac{2}{20}$
$C_{(2,1)} = \frac{8}{20}$
$C_{(1,2)} = \frac{6}{20}$
$C_{(2,2)} = \frac{4}{20}$

Figure 5.1.: Example graph G of size $N = 20$, sampled from a stochastic block model with two clusters. Each node has one of two states, state 1 is indicated by blue color and state 2 by green color. A node is assigned to class k if it is located in cluster k, $k = 1, 2$. The collective variable $C(x)$ measures the shares of the two states separately for each cluster. Reprinted from [1].

3. If nodes differ by their transition rate matrices Q_i^G, i.e., the population is heterogeneous, it is reasonable to assign them to different classes. As an example, there could be very active nodes (large transition rates) and rather inactive nodes (small or zero transition rates, often referred to as "zealots" in the literature [141, 139, 142]). The case of such a heterogeneous population is discussed further in section 5.2.2.

Due to the dynamics, a node with extended state (m, k) may transition to any other state $n \neq m$, such that after the transition it has extended state (n, k). This transition will be referred to as $(m, k) \to n$ and there are $MK(M-1)$ transitions in total. Each transition has an associated state-change vector $v_{(m,k)\to n} \in \mathbb{Z}^{MK}$ and an associated propensity function $\alpha^G_{(m,k)\to n} : [M]^N \to \mathbb{R}_{\geq 0}$. The state-change vector

$$v_{(m,k)\to n} := e_{(n,k)} - e_{(m,k)}, \tag{5.3}$$

where $e_{(m,k)}$ denotes the (m, k)-th unit-vector, describes the changes in extended state populations due to the transition $(m, k) \to n$, i.e., the number of nodes in extended state (m, k) decreases by one and the number of nodes in extended state (n, k) increases by one. The propensity function $\alpha^G_{(m,k)\to n}$ measures the cumulative rate of the transition $(m, k) \to n$, i.e., the sum of the transition rates of all nodes with extended state (m, k) to state n

$$\alpha^G_{(m,k)\to n}(x) := \sum_{i\in[N]:(x_i,s_i)=(m,k)} \left(Q_i^G(x)\right)_{m,n}. \tag{5.4}$$

In the following, the summation over all $MK(M-1)$ transitions will be abbreviated with the symbol

$$\sum_{(m,k)\to n} := \sum_{m=1}^{M} \sum_{k=1}^{K} \sum_{\substack{n=1\\ n\neq m}}^{M}, \tag{5.5}$$

and analogously the symbol "$\max_{(m,k)\to n}$" is used for the maximum over all transitions.

The remainder of this chapter is structured as follows. Section 5.1 contains the main theorem for the convergence to a mean-field limit and some considerations regarding the rate of convergence. In section 5.2 this main theorem is applied to the continuous-time noisy voter model to show convergence for several graph models, i.e., Erdős–Rényi random graphs, the stochastic block model, and random regular graphs, including bounds for the specific parameters of the graph models. Moreover, the case of a heterogeneous population, where nodes have varying transition rate functions Q_i, is discussed. In section 5.3 the above-mentioned mean-field limit is augmented with a stochastic term. The resulting SDE is able to approximate the dynamics well even for medium-sized populations for which the mean-field limit is too inaccurate. Finally, in section 5.4 the large population limits of leader-follower dynamics are discussed. In such systems, a few very influential agents called *leaders* affect a large mass of ordinary agents called *followers*.

Sections 5.1 and 5.2 are largely adopted from the author's publication [1].

5.1. Convergence conditions

In this section conditions for the convergence to a mean-field limit are derived. Before proving the main theorem 5.2, which precisely formulates these conditions, an intuitive derivation is presented.

The conditions of the main theorem state that it has to be possible to choose classes such that the collective variables (5.2) are *good* as discussed in section 2.3, that is, they capture the most important dynamical information. This translates to the condition that the propensities (5.4) of the transitions can be approximated well by using only the macroscopic information $C(x)$ of the state, i.e., there exist propensity functions $\tilde{\alpha}_{(m,k)\to n} : \mathbb{R}^{MK} \to \mathbb{R}$ such that

$$\frac{1}{N}\alpha^{G}_{(m,k)\to n}(x) \approx \tilde{\alpha}_{(m,k)\to n}\big(C(x)\big) \quad \forall x \in [M]^N. \tag{5.6}$$

It is assumed that all $\tilde{\alpha}_{(m,k)\to n}$ are Lipschitz continuous. The existence of such an appropriate choice of classes and reduced propensity functions $\tilde{\alpha}_{(m,k)\to n}$ for a given dynamical system on a certain network is not clear, and finding them is no trivial task. However, if classes and propensity functions can be found such that the approximation (5.6) becomes exact in the large population limit, then there exists a mean-field ODE describing the projected system state $C(x)$, which is shown in theorem 5.2.

In order to specify the large population limit, consider a sequence of random graphs $(\boldsymbol{G}_\ell)_{\ell\in\mathbb{N}}$, such that $\boldsymbol{G}_\ell$ has N_ℓ nodes and the sequence $(N_\ell)_{\ell\in\mathbb{N}}$ is strictly increasing. Furthermore, let $s_i^\ell \in [K]$ denote the class of node i of the random graph $\boldsymbol{G}_\ell$ and define the collective variables

$$C^\ell_{(m,k)}(x) := \frac{\#\{i \in [N_\ell] : (x_i, s_i^\ell) = (m,k)\}}{N_\ell}. \tag{5.7}$$

Let $\boldsymbol{x}^\ell(t)$ denote the stochastic jump process on the random graph $\boldsymbol{G}_\ell$, and

$$\boldsymbol{c}^\ell(t) := C^\ell(\boldsymbol{x}^\ell(t)) \tag{5.8}$$

the projected process. In order to quantify the approximation (5.6), define the difference

$$\Delta^{\boldsymbol{G}_\ell}(x) := \max_{(m,k)\to n} \left| \frac{1}{N_\ell}\alpha^{\boldsymbol{G}_\ell}_{(m,k)\to n}(x) - \tilde{\alpha}_{(m,k)\to n}\big(C^\ell(x)\big) \right| \tag{5.9}$$

for $x \in [M]^{N_\ell}$. Moreover, it is assumed that the transition rate matrices $Q_i^G(x)$ have bounded entries, i.e., there is a bound $B > 0$, such that for any graph G with any number of nodes

$$Q_i^G(x) < B \quad \text{element-wise for any system state } x \text{ and all } i \in [N]. \tag{5.10}$$

If this assumption of bounded rates would not be satisfied, the dynamics could become "infinitely fast" such that a meaningful large population limit can not be defined. For models where the rates Q_i^G increase with the total number of nodes or with the (average) degree of nodes, an appropriate rescaling factor may be applied to the rates to achieve boundedness.

The above-mentioned mean-field ODE, which will be called the *mean-field equation* (MFE) in the following, is then given by

$$\frac{\mathrm{d}}{\mathrm{d}t}c(t) = \sum_{(m,k)\to n} \tilde{\alpha}_{(m,k)\to n}(c(t))\; v_{(m,k)\to n} =: F\big(c(t)\big), \tag{5.11}$$

where $c : \mathbb{R} \to \mathbb{R}^{MK}$ describes the trajectory of the shares. The infinitesimal change in c is characterized by the propensities of each transition $\tilde{\alpha}_{(m,k)\to n}$ times their effect on the extended state populations $v_{(m,k)\to n}$. Due to the assumption that the $\tilde{\alpha}_{(m,k)\to n}$ are Lipschitz continuous, the mean-field ODE has a unique solution, given initial condition $c(0) = c_0$.

The following main theorem of the section is inspired by work about the concentration of Markov processes by Kurtz [143], which can be applied to a sequence of complete graphs increasing in size. The proof below generalizes the proof of Kurtz's result presented in [40, Thm. A.14] to the framework of random graphs, and is relying on combining the law of large numbers and Gronwall's lemma.

Theorem 5.2. *Assume that for all $\varepsilon > 0$ there exists a function $f_\varepsilon : \mathbb{N} \to \mathbb{R}_{\geq 0}$ with $\lim_{\ell\to\infty} f_\varepsilon(\ell) = 0$ such that*

$$\forall \ell \in \mathbb{N}: \; \mathbb{P}\Big(\max_{x\in[M]^{N_\ell}} \Delta^{\boldsymbol{G}_\ell}(x) \geq \varepsilon \Big) \leq f_\varepsilon(\ell). \tag{5.12}$$

Furthermore, let there be initial conditions $\boldsymbol{x}^\ell(0)$, such that $\boldsymbol{c}^\ell(0) \xrightarrow{\mathrm{P}} c_0 \in \mathbb{R}^{MK}$ as $\ell \to \infty$. Let $c(\cdot)$ denote the solution of the mean-field equation (5.11) with initial condition $c(0) = c_0$. Then

$$\forall t \geq 0: \; \sup_{0\leq s\leq t} \big\| \boldsymbol{c}^\ell(s) - c(s) \big\| \xrightarrow[\ell\to\infty]{\mathrm{P}} 0. \tag{5.13}$$

Proof. Let $\big(\boldsymbol{P}_{(m,k)\to n}(t)\big)_{m,k,n}$ denote independent unit-rate Poisson processes. Then the projected process can be written as (see [40, section 1.2])

$$\boldsymbol{c}^\ell(t) = \boldsymbol{c}^\ell(0) + \sum_{(m,k)\to n} \boldsymbol{P}_{(m,k)\to n}\Big(\int_0^t \alpha^{\boldsymbol{G}_\ell}_{(m,k)\to n}(\boldsymbol{x}^\ell(s))\,\mathrm{d}s \Big) \frac{v_{(m,k)\to n}}{N_\ell}. \tag{5.14}$$

Defining the centered Poisson processes $\tilde{\boldsymbol{P}}_{(m,k)\to n}(t) := \boldsymbol{P}_{(m,k)\to n}(t) - t$ yields

$$\begin{aligned}\boldsymbol{c}^\ell(t) &= \boldsymbol{c}^\ell(0) \\ &\quad + \underbrace{\sum_{(m,k)\to n} \tilde{\boldsymbol{P}}_{(m,k)\to n}\Big(\int_0^t \alpha^{\boldsymbol{G}_\ell}_{(m,k)\to n}(\boldsymbol{x}^\ell(s))\,\mathrm{d}s \Big) \frac{v_{(m,k)\to n}}{N_\ell}}_{=:\boldsymbol{\delta}_\ell(t)} + \int_0^t \boldsymbol{F}_\ell(\boldsymbol{x}^\ell(s))\mathrm{d}s,\end{aligned} \tag{5.15}$$

where

$$\boldsymbol{F}_\ell(x) := \sum_{(m,k)\to n} \alpha^{\boldsymbol{G}_\ell}_{(m,k)\to n}(x)\frac{v_{(m,k)\to n}}{N_\ell}. \tag{5.16}$$

Note that due to the assumption (5.10) that transition rates are bounded by $B > 0$, it follows

$$\alpha^{\boldsymbol{G}_\ell}_{(m,k)\to n}(x) = \sum_{i:(x_i,s_i)=(m,k)} \big(Q^{\boldsymbol{G}_\ell}_i(x)\big)_{m,n} \leq N_\ell B \tag{5.17}$$

and thus

$$\begin{aligned}\hat{\boldsymbol{\delta}}_\ell(t) &:= \sup_{0\leq s\leq t} \|\boldsymbol{\delta}_\ell(s)\| && (5.18)\\ &\leq \sum_{(m,k)\to n} \sup_{0\leq s\leq t}\Big|\frac{1}{N_\ell}\tilde{\boldsymbol{P}}_{(m,k)\to n}(sN_\ell B)\Big|\|v_{(m,k)\to n}\|. && (5.19)\end{aligned}$$

By the law of large numbers, one can show that (see for example [144, Theorem 1.2])

$$\sup_{0\leq s\leq t}\big|\frac{1}{N_\ell}\tilde{\boldsymbol{P}}_{(m,k)\to n}(sN_\ell B)\big| \xrightarrow{\mathrm{p}} 0 \quad \text{as } \ell\to\infty \tag{5.20}$$

and hence

$$\forall t:\ \hat{\boldsymbol{\delta}}_\ell(t) \xrightarrow[\ell\to\infty]{\mathrm{p}} 0. \tag{5.21}$$

Furthermore, consider that

$$\begin{aligned}&\Big\|\int_0^t \boldsymbol{F}_\ell(\boldsymbol{x}^\ell(s)) - F(\boldsymbol{c}^\ell(s))ds\Big\|\\ &\quad\leq \int_0^t \underbrace{\sum_{(m,k)\to n}\Big|\frac{1}{N_\ell}\alpha^{\boldsymbol{G}_\ell}_{(m,k)\to n}(\boldsymbol{x}^\ell(s)) - \tilde{\alpha}_{(m,k)\to n}(\boldsymbol{c}^\ell(s))\Big|\|v_{(m,k)\to n}\|}_{=:\boldsymbol{z}^\ell(s)}\,\mathrm{d}s =: \tilde{\boldsymbol{\delta}}_\ell(t) \quad (5.22)\end{aligned}$$

and

$$\boldsymbol{z}^\ell(s) \leq \sum_{(m,k)\to n}\Delta^{\boldsymbol{G}_\ell}\big(\boldsymbol{x}^\ell(s)\big)\bar{v} = MK(M-1)\ \Delta^{\boldsymbol{G}_\ell}\big(\boldsymbol{x}^\ell(s)\big)\bar{v}, \tag{5.23}$$

where $\bar{v} := \max_{(m,k)\to n}\|v_{(m,k)\to n}\|$. Let $\varepsilon > 0$ and define $\tilde{\varepsilon} := \frac{\varepsilon}{MK(M-1)\bar{v}}$. Then it follows that

$$\begin{aligned}\mathbb{P}(\boldsymbol{z}^\ell(s)\geq\varepsilon) &\leq \mathbb{P}\Big(\Delta^{\boldsymbol{G}_\ell}\big(\boldsymbol{x}^\ell(s)\big)\geq\tilde{\varepsilon}\Big) && (5.24)\\ &\leq \mathbb{P}\Big(\max_{x\in[M]^{N_\ell}}\Delta^{\boldsymbol{G}_\ell}(x)\geq\tilde{\varepsilon}\Big) && (5.25)\\ &\leq f_{\tilde{\varepsilon}}(\ell)\xrightarrow{\ell\to\infty} 0. && (5.26)\end{aligned}$$

Hence, from lemma A.2 it follows that

$$\tilde{\boldsymbol{\delta}}_\ell(t) = \int_0^t \boldsymbol{z}^\ell(s)\mathrm{d}s \xrightarrow[\ell\to\infty]{\mathrm{p}} 0. \tag{5.27}$$

Now, writing $c(t) = c(0) + \int_0^t F(c(s))\mathrm{d}s$ and $\boldsymbol{c}^\ell(t)$ as in (5.15) yields

$$\begin{aligned} \|\boldsymbol{c}^\ell(t) - c(t)\| &= \left\| \boldsymbol{c}^\ell(0) - c(0) + \boldsymbol{\delta}_\ell(t) + \int_0^t \boldsymbol{F}_\ell(\boldsymbol{x}^\ell(s)) - F(c(s))\mathrm{d}s \right\| && (5.28) \\ &\leq \|\boldsymbol{c}^\ell(0) - c(0)\| + \|\boldsymbol{\delta}_\ell(t)\| + \left\| \int_0^t \boldsymbol{F}_\ell(\boldsymbol{x}^\ell(s)) - F(\boldsymbol{c}^\ell(s))\mathrm{d}s \right\| && (5.29) \\ &\qquad + \left\| \int_0^t F(\boldsymbol{c}^\ell(s)) - F(c(s))\mathrm{d}s \right\| \\ &\leq \|\boldsymbol{c}^\ell(0) - c(0)\| + \hat{\boldsymbol{\delta}}_\ell(t) + \tilde{\boldsymbol{\delta}}_\ell(t) + L \int_0^t \|\boldsymbol{c}^\ell(s) - c(s)\|\mathrm{d}s, && (5.30) \end{aligned}$$

where L denotes the Lipschitz constant of F. (F is Lipschitz continuous because of the assumption that all $\tilde{\alpha}_{(m,k)\to n}$ are.) Note that $\hat{\boldsymbol{\delta}}_\ell(t)$ and $\tilde{\boldsymbol{\delta}}_\ell(t)$ are monotonically increasing in t. Thus, by the Gronwall lemma it follows that

$$\|\boldsymbol{c}^\ell(t) - c(t)\| \leq \left(\|\boldsymbol{c}^\ell(0) - c(0)\| + \hat{\boldsymbol{\delta}}_\ell(t) + \tilde{\boldsymbol{\delta}}_\ell(t) \right) \exp(Lt) \xrightarrow[\ell\to\infty]{\mathrm{P}} 0 \tag{5.31}$$

due to (5.21) and (5.27). Because of the monotonicity of the above bound, the theorem follows. □

Remark 5.3. The case of a sequence of deterministic graphs $(G_\ell)_{\ell\in\mathbb{N}}$ (as opposed to random graphs) is also contained in the previous theorem. In this case, the condition of the theorem collapses to

$$\max_{x\in[M]^{N_\ell}} \Delta^{G_\ell}(x) \overset{\ell\to\infty}{\longrightarrow} 0. \tag{5.32}$$

Theorem 5.2 provides conditions that guarantee the convergence of the projected process $\boldsymbol{c}^\ell(t)$ to a mean field limit. As mentioned before, this projected process contains both the randomness of the graph and of the dynamics, and thus theorem 5.2 is called the *annealed* result. The *quenched* result on the other hand describes the setting that the realization of the random graph is fixed and only the randomness of the dynamics is considered.

Corollary 5.4 (Quenched result)**.** *If the function f_ε from theorem 5.2 additionally satisfies*

$$\forall \varepsilon > 0: \sum_{\ell=1}^{\infty} f_\varepsilon(\ell) < \infty, \tag{5.33}$$

the convergence to the mean-field limit holds for almost all realizations of the sequence of random graphs. More precisely, if $\boldsymbol{c}^\ell_G(s)$ denotes the stochastic process given by the dynamics on a fixed graph G, then for almost all realizations $(G_\ell)_\ell$ of the sequence of random graphs $(\boldsymbol{G}_\ell)_\ell$ it follows

$$\forall t \geq 0: \sup_{0\leq s\leq t} \left\| \boldsymbol{c}^\ell_{G_\ell}(s) - c(s) \right\| \xrightarrow[\ell\to\infty]{\mathrm{P}} 0. \tag{5.34}$$

Proof. Due to the Borel–Cantelli lemma, with probability 1 the event

$$\left\{ \max_{x\in[M]^{N_\ell}} \Delta^{\boldsymbol{G}_\ell}(x) \geq \varepsilon \right\} \tag{5.35}$$

only occurs for finitely many ℓ. As the choice of $\varepsilon > 0$ was arbitrary, it follows

$$\mathbb{P}\left(\max_{x \in [M]^{N_\ell}} \Delta^{\boldsymbol{G}_\ell}(x) \xrightarrow[\ell \to \infty]{} 0 \right) = 1. \tag{5.36}$$

Hence, remark 5.3 can be applied for almost all realizations of the sequence of random graphs. □

Rate of convergence. The proof of theorem 5.2, in particular the bound (5.31)

$$\sup_{0 \le s \le t} \|\boldsymbol{c}^\ell(s) - c(s)\| \le \Big(\|\boldsymbol{c}^\ell(0) - c(0)\| + \hat{\boldsymbol{\delta}}_\ell(t) + \tilde{\boldsymbol{\delta}}_\ell(t) \Big) \exp(Lt)$$

allows the specification of the rate of convergence of the stochastic process $\boldsymbol{c}^\ell(t)$ to the mean-field solution $c(t)$.

First of all, note that the factor $\exp(Lt)$ implies that the deviation between the stochastic process $\boldsymbol{c}^\ell(t)$ and the mean-field solution $c(t)$ may increase over time for a fixed ℓ. The rate L of this deterioration is proportional to the Lipschitz constants of the propensities $\tilde{\alpha}_{(m,k)\to n}$. Hence, for practical purposes, if a good match between model and mean-field solution is desired for a longer time, the network size N_ℓ may have to be increased substantially. However, since the factor $\exp(Lt)$ does not depend on ℓ, it is not relevant for the rate of convergence. Moreover, the first term inside the brackets, $\|\boldsymbol{c}^\ell(0) - c(0)\|$, merely quantifies the difference of initial conditions and is typically not a limiting factor. For example, for any target initial condition $c_0 \in [0,1]^{MK}$, there exist (deterministic) initial conditions $\boldsymbol{c}^\ell(0)$ on the graph with an error $\|\boldsymbol{c}^\ell(0) - c_0\| = \mathcal{O}(N_\ell^{-1})$. Hence, the two relevant quantities for the rate of convergence are $\hat{\boldsymbol{\delta}}_\ell(t)$ and $\tilde{\boldsymbol{\delta}}_\ell(t)$.

The convergence of the term $\hat{\boldsymbol{\delta}}_\ell(t)$ to 0 is essentially given by the law of large numbers, applied to normalized and centered Poisson processes. Thus, a rate of convergence of $\sqrt{N_\ell}^{-1}$ is dictated by the central limit theorem, see the following lemma.

Lemma 5.5. *Let $\hat{\boldsymbol{\delta}}_\ell(t)$ be as defined in the proof of theorem 5.2. Then*

$$\mathbb{E}[\hat{\boldsymbol{\delta}}_\ell(t)] = \mathcal{O}\big(\sqrt{N_\ell}^{-1}\big). \tag{5.37}$$

Proof. Neglecting constant factors, it follows from (5.19) that $\mathbb{E}[\hat{\boldsymbol{\delta}}_\ell(t)]$ is bounded by

$$\mathbb{E}\left[\sup_{0 \le s \le t} \left| \frac{1}{N_\ell} \tilde{\boldsymbol{P}}(sN_\ell B) \right| \right], \tag{5.38}$$

where $\tilde{\boldsymbol{P}}$ is a centered Poisson process and $B > 0$ is the bound of the transition rates, see (5.10). One can show [144, Lemma 1.3] that the centered Poisson process is approximated well by Brownian motion $\boldsymbol{W}$, i.e., for all $\ell \in \mathbb{N}$

$$\boldsymbol{\Gamma} := \sup_{t \ge 0} \frac{|\tilde{\boldsymbol{P}}(tN_\ell B) - \boldsymbol{W}(tN_\ell B)|}{\log(\max(2, tN_\ell B))} < \infty \quad \text{a.s.}, \qquad \mathbb{E}[\boldsymbol{\Gamma}] < \infty, \tag{5.39}$$

from which follows (assuming ℓ large enough so that $tN_\ell B \ge 2$)

$$\sup_{0 \le s \le t} \left| \tilde{\boldsymbol{P}}(sN_\ell B) - \sqrt{N_\ell}\, \boldsymbol{W}(sB) \right| \le \boldsymbol{\Gamma} \log(tN_\ell B). \tag{5.40}$$

This implies

$$\sup_{0\le s\le t}\left|\frac{1}{N_\ell}\tilde{\boldsymbol{P}}(sN_\ell B)\right| \le \boldsymbol{\Gamma}\frac{\log(tN_\ell B)}{N_\ell} + \sup_{0\le s\le t}\left|\frac{1}{\sqrt{N_\ell}}\boldsymbol{W}(sB)\right| \tag{5.41}$$

and

$$\mathbb{E}\left[\sup_{0\le s\le t}\left|\frac{1}{N_\ell}\tilde{\boldsymbol{P}}(sN_\ell B)\right|\right] \le \mathbb{E}[\boldsymbol{\Gamma}]\frac{\log(tN_\ell B)}{N_\ell} + \frac{1}{\sqrt{N_\ell}}\mathbb{E}\left[\sup_{0\le s\le t}|\boldsymbol{W}(sB)|\right], \tag{5.42}$$

from which the claim $\mathbb{E}[\hat{\boldsymbol{\delta}}_\ell(t)] = \mathcal{O}(\sqrt{N_\ell}^{-1})$ follows. □

The other relevant term, $\tilde{\boldsymbol{\delta}}_\ell(t)$, captures the influence of the random graph. The smaller the functions f_ε in theorem 5.2 can be chosen, the faster is the convergence of $\tilde{\boldsymbol{\delta}}_\ell(t)$. More precisely, the rate of convergence depends on how fast ε can be sent to 0 while ℓ goes to infinity, such that $f_\varepsilon(\ell)$ still falls sufficiently quickly.

Lemma 5.6. *Let $f_\varepsilon(\ell)$ and $\tilde{\boldsymbol{\delta}}_\ell(t)$ be as defined in theorem 5.2. Assume that there exists a function $\varepsilon(\ell) > 0$ with $\lim_{\ell\to\infty}\varepsilon(\ell) = 0$ such that $f_{\varepsilon(\ell)}(\ell) = \mathcal{O}(\varepsilon(\ell))$ as $\ell\to\infty$. Then*

$$\mathbb{E}\big[\tilde{\boldsymbol{\delta}}_\ell(t)\big] = \mathcal{O}(\varepsilon(\ell)) \quad \text{as } \ell\to\infty.$$

Proof. By applying Tonelli's theorem to interchange integral and expected value, it follows that

$$\mathbb{E}\big[\tilde{\boldsymbol{\delta}}_\ell(t)\big] = \mathbb{E}\Big[\int_0^t \boldsymbol{z}^\ell(s)\mathrm{d}s\Big] = \int_0^t \mathbb{E}\big[\boldsymbol{z}^\ell(s)\big]\mathrm{d}s. \tag{5.43}$$

From the proof of lemma A.2 it can be seen that $\mathbb{E}[\boldsymbol{z}^\ell(s)] \le \varepsilon + f_\varepsilon(\ell)$ for all $\varepsilon > 0$, which implies

$$\mathbb{E}\big[\tilde{\boldsymbol{\delta}}_\ell(t)\big] \le t\,\varepsilon + t\,f_\varepsilon(\ell). \tag{5.44}$$

Inserting the function $\varepsilon(\ell)$ yields

$$\mathbb{E}\big[\tilde{\boldsymbol{\delta}}_\ell(t)\big] \le t\,\varepsilon(\ell) + t\,f_{\varepsilon(\ell)}(\ell) = \mathcal{O}(\varepsilon(\ell)), \tag{5.45}$$

which concludes the proof. □

The overall rate of convergence is summarized in the following proposition.

Proposition 5.7. *Assuming that initial conditions are chosen appropriately and that the function $\varepsilon(\ell)$ from lemma 5.6 exists, the rate of convergence of the stochastic process $\boldsymbol{c}^\ell(t)$ to the mean-field solution $c(t)$, as discussed in theorem 5.2, is the slower one of $\sqrt{N_\ell}^{-1}$ and $\varepsilon(\ell)$, i.e.,*

$$\mathbb{E}\Big[\sup_{0\le s\le t}\|\boldsymbol{c}^\ell(s) - c(s)\|\Big] = \mathcal{O}\big(\sqrt{N_\ell}^{-1} + \varepsilon(\ell)\big) \quad \text{as } \ell\to\infty. \tag{5.46}$$

5.2. Application to the voter model

In this section large population limits of the continuous-time noisy voter model (CNVM) are analyzed for a selection of popular random graphs by verifying the conditions for the existence of the mean-field limit presented in theorem 5.2. First, Erdős–Rényi random graphs are discussed in section 5.2.1 and a lower bound for the edge density is derived that guarantees convergence to the mean-field limit. These results are modified in section 5.2.2 to incorporate a heterogeneous population where the agents may have varying transition rates. Graphs generated by the stochastic block model are examined in section 5.2.3. Finally, in section 5.2.4 uniformly random regular graphs are discussed and it is investigated how large the degree of nodes has to be for convergence to the mean-field limit.

To conform to the assumption (5.10) of bounded transition rates, the CNVM will be considered with parameter $\alpha = 1$, i.e., the rate at which an agent transitions from opinion m to n is given by

$$\left(Q_i^G(x)\right)_{m,n} \coloneqq r_{m,n} \frac{d_{i,n}^G(x)}{d_i^G} + \tilde{r}_{m,n}, \tag{5.47}$$

where $d_{i,n}^G(x)$ denotes the number of neighbors of node i with opinion n, d_i^G is the degree of node i, and $r_{m,n}, \tilde{r}_{m,n} \geq 0$ are model parameters. Again, the superscript G is used to denote the dependence on the underlying graph G.

5.2.1. Erdős–Rényi random graphs

Recall that the Erdős–Rényi (ER) random graph $\boldsymbol{G}_{N,p}$, also called binomial random graph, is defined as the random graph where each possible edge appears independently with probability $p > 0$, see section 3.1.1 for more details. In the following it is implicitly assumed that the edge probability $p = p(N)$ may depend on the number of nodes N. It is especially interesting to investigate convergence to the mean-field limit depending on the asymptotic behavior of p, i.e., depending on how fast p converges to 0 as $N \to \infty$.

In this section it is shown that the CNVM on ER random graphs converges to a mean-field limit with respect to the shares of each opinion, provided p is large enough. Thus, there is only one class $K = 1$ and the extended states $(x_i, s_i) = (x_i, 1)$ collapse to just the states x_i. For easier notation, the transition $(m, 1) \to n$ is referred to as $m \to n$. Moreover, the sequence of random graphs $\left(\boldsymbol{G}_{N,p}\right)_{N \in \mathbb{N}}$ is considered, such that the index N is used instead of ℓ from the previous section.

Heuristic derivation of the mean-field equation. In the setting described above, consider the propensity functions for a fixed graph G

$$\alpha_{m \to n}^G(x) = \sum_{i : x_i = m} \left(Q_i^G(x)\right)_{m,n} \tag{5.48}$$

$$= \sum_{i : x_i = m} \left(r_{m,n} \frac{d_{i,n}^G(x)}{d_i^G} + \tilde{r}_{m,n}\right) \tag{5.49}$$

describing the cumulative rate at which an opinion change $m \to n$ occurs in the entire system. Because of the homogeneous nature of an ER random graph $\boldsymbol{G}$, the share of agents of opinion n in the neighborhood of agent i, $d_{i,n}^{\boldsymbol{G}}(x)/d_i^{\boldsymbol{G}}$, is expected to be approximately equal to the share of opinion n in the whole system, $C_n(x) =: c_n$. This leads to the following choice of reduced propensity function $\tilde{\alpha}_{m\to n}$:

$$\frac{1}{N}\alpha_{m\to n}^{\boldsymbol{G}}(x) \approx \frac{1}{N}\sum_{i:x_i=m}\Big(r_{m,n}c_n + \tilde{r}_{m,n}\Big) \tag{5.50}$$

$$= c_m\big(r_{m,n}c_n + \tilde{r}_{m,n}\big) =: \tilde{\alpha}_{m\to n}(c). \tag{5.51}$$

Inserting into (5.11) yields the associated mean-field ODE

$$\frac{\mathrm{d}}{\mathrm{d}t}c(t) = \sum_{m\neq n} c_m(t)\big(r_{m,n}c_n(t) + \tilde{r}_{m,n}\big)(e_n - e_m), \tag{5.52}$$

where the sum is over all pairs $(m,n) \in [M]\times[M]$ with $m \neq n$. Not surprisingly, this is the same equation as was derived in section 2.4.1 by considering the general principles of mean-field theory.

Auxiliary concentration results. To show that the conditions of theorem 5.2 are indeed fulfilled for the above choice of propensity functions $\tilde{\alpha}_{m\to n}$, the following auxiliary results are useful.

Lemma 5.8. *Let $\boldsymbol{G} = \boldsymbol{G}_{N,p}$ denote the ER random graph and $x \in [M]^N$ an arbitrary but fixed state. Define the random variable $\boldsymbol{E}_{m,n}$ as the number of edges between nodes of opinion m and nodes of opinion $n \neq m$ in $\boldsymbol{G}$, according to x, i.e., $\boldsymbol{E}_{m,n} := \sum_{i:x_i=m} d_{i,n}^{\boldsymbol{G}}(x)$. Then the concentration inequality*

$$\mathbb{P}\Big(\frac{1}{N^2p}|\boldsymbol{E}_{m,n} - c_mc_nN^2p| \geq \varepsilon\Big) \leq 2\exp\Big(-\frac{\varepsilon^2N^2p}{3}\Big), \tag{5.53}$$

where $c := C(x)$, holds for all $\varepsilon > 0$.

Proof. There are $u := c_mc_nN^2$ possible edges between m-opinion and n-opinion nodes. As every edge in $\boldsymbol{G}_{N,p}$ is present with probability p independently of all other edges, it follows that $\boldsymbol{E}_{m,n}$ is binomial distributed with u trials and success probability p. In particular, this implies $\mathbb{E}[\boldsymbol{E}_{m,n}] = up = c_mc_nN^2p$, and applying the Chernoff bound (see lemma A.1) yields

$$\mathbb{P}(|\boldsymbol{E}_{m,n} - c_mc_nN^2p| \geq \varepsilon) \leq 2\exp\Big(-\frac{\varepsilon^2}{3up}\Big) \tag{5.54}$$

$$\leq 2\exp\Big(-\frac{\varepsilon^2}{3N^2p}\Big), \tag{5.55}$$

where the last inequality follows from $u \leq N^2$. □

Furthermore, recall that in $\boldsymbol{G}_{N,p}$ the node degrees concentrate around the mean degree. The following modified version of lemma 3.3 will be useful later.

Lemma 5.9. *For all $\varepsilon > 0$ and $i \in [N]$ it holds that*

$$\mathbb{P}\Big(\big|d_i^{\boldsymbol{G}_{N,p}} - Np\big| \geq \varepsilon Np\Big) \leq 2\exp\Big(-\frac{\varepsilon^2Np}{3} + \frac{2\varepsilon}{3}\Big). \tag{5.56}$$

Proof. Application of the Chernoff bound (see lemma A.1) yields

$$\mathbb{P}\left(\left|d_i^{\boldsymbol{G}_{N,p}} - Np\right| \geq \varepsilon Np\right) \leq \mathbb{P}\left(\left|d_i^{\boldsymbol{G}_{N,p}} - (N-1)p\right| + p \geq \varepsilon Np\right) \tag{5.57}$$

$$\leq 2\exp\Big(-\frac{(\varepsilon Np - p)^2}{3(N-1)p}\Big) \tag{5.58}$$

$$\leq 2\exp\Big(-\frac{\varepsilon^2N^2p^2 + 2\varepsilon Np^2}{3Np}\Big), \tag{5.59}$$

from which the lemma follows. □

Large population limit. Using the above auxiliary results the conditions of theorem 5.2 are verified in the following proposition.

Proposition 5.10. *Let $\boldsymbol{G}_{N,p}$ denote the ER random graph and $\hat{r} := \max_{m\neq n} r_{m,n}$. For all $\varepsilon \in (0,\hat{r})$ it holds that*

$$\forall N \in \mathbb{N}: \ \mathbb{P}\Big(\max_{x\in[M]^N} \Delta^{\boldsymbol{G}_{N,p}}(x) \geq \varepsilon\Big) \leq f_\varepsilon(N), \tag{5.60}$$

where

$$f_\varepsilon(N) := 4M^{N+2}\exp\Big(-\frac{1}{12}N^2p\Big(\frac{\varepsilon}{\hat{r}} - \frac{\varepsilon^2}{\hat{r}^2}\Big)^2\Big) + 2N\exp\Big(-N\frac{\varepsilon^2 p}{12\hat{r}} + \frac{\varepsilon}{3\hat{r}}\Big). \tag{5.61}$$

Proof. Fix any $N \in \mathbb{N}$ and denote $\boldsymbol{G} := \boldsymbol{G}_{N,p}$. By inserting the propensity functions (5.49) and (5.51), it follows that

$$\Delta^{\boldsymbol{G}}(x) = \max_{m\neq n} r_{m,n}\Big|\frac{1}{N}\sum_{i:x_i=m}\frac{d_{i,n}^{\boldsymbol{G}}(x)}{d_i^{\boldsymbol{G}}} - C_m(x)C_n(x)\Big|. \tag{5.62}$$

Let $\varepsilon \in (0,\hat{r})$, $\delta \in (0,1)$, and define the events

$$\begin{aligned}\mathcal{A} &:= \Big\{\max_{x\in[M]^N} \Delta^{\boldsymbol{G}_N}(x) \geq \varepsilon\Big\} \\ &= \Big\{\max_{x\in[M]^N}\max_{m\neq n} r_{m,n}\Big|\frac{1}{N}\sum_{i:x_i=m}\frac{d_{i,n}^{\boldsymbol{G}}(x)}{d_i^{\boldsymbol{G}}} - C_m(x)C_n(x)\Big| \geq \varepsilon\Big\}, \qquad (5.63)\\ \mathcal{B} &:= \Big\{\forall i : (1-\delta)Np \leq d_i^{\boldsymbol{G}} \leq (1+\delta)Np\Big\}. \qquad (5.64)\end{aligned}$$

From the concentration of degrees in $\boldsymbol{G}$, see lemma 5.9, and the union bound it follows that

$$\mathbb{P}(\mathcal{B}^C) \leq 2N\exp\Big(\frac{1}{3}(-\delta^2Np + 2\delta)\Big), \tag{5.65}$$

where $\mathcal{B}^C$ denotes the complement of $\mathcal{B}$. The goal is now to derive a bound for $\mathbb{P}(\mathcal{A})$ (from which the proposition follows) by showing that $\mathbb{P}(\mathcal{A}\cap\mathcal{B})$ is small and combining that with (5.65). Define the number of m to n edges $\boldsymbol{E}_{m,n}^x := \sum_{i:x_i=m} d_{i,n}^{\boldsymbol{G}}(x)$ like in lemma 5.8. For any fixed state $x \in [M]^N$ and any opinions $m \neq n$, it follows that

$$\begin{aligned}
&\mathbb{P}\Big(r_{m,n}\Big|\frac{1}{N}\sum_{i:x_i=m}\frac{d^{\boldsymbol{G}}_{i,n}(x)}{(1\pm\delta)Np} - C_m(x)C_n(x)\Big| \geq \varepsilon\Big) && (5.66)\\
&= \mathbb{P}\Big(\frac{1}{(1\pm\delta)N^2p} r_{m,n}\big|\boldsymbol{E}^x_{m,n} - C_m(x)C_n(x)(1\pm\delta)N^2p\big| \geq \varepsilon\Big) && (5.67)\\
&\leq \mathbb{P}\Big(\big|\boldsymbol{E}^x_{m,n} - C_m(x)C_n(x)N^2p\big| + \underbrace{C_m(x)C_n(x)}_{\leq 1}\delta N^2p \geq r^{-1}_{m,n}(1\pm\delta)N^2p\varepsilon\Big) && (5.68)\\
&\leq \mathbb{P}\Big(\big|\boldsymbol{E}^x_{m,n} - C_m(x)C_n(x)N^2p\big| \geq r^{-1}_{m,n}(1\pm\delta)N^2p\varepsilon - \delta N^2p\Big) && (5.69)\\
&\leq \mathbb{P}\Big(\big|\boldsymbol{E}^x_{m,n} - C_m(x)C_n(x)N^2p\big| \geq N^2p(\hat{r}^{-1}\varepsilon - \hat{r}^{-1}\varepsilon\delta - \delta)\Big). && (5.70)
\end{aligned}$$

This also holds after applying the maximum over states x:

$$\begin{aligned}
&\mathbb{P}\Big(\max_{x\in[M]^N}\max_{m\neq n} r_{m,n}\Big|\frac{1}{N}\sum_{i:x_i=m}\frac{d^{\boldsymbol{G}}_{i,n}(x)}{(1\pm\delta)Np} - C_m(x)C_n(x)\Big| \geq \varepsilon\Big) && (5.71)\\
&\leq \mathbb{P}\Big(\max_{x\in[M]^N}\max_{m\neq n}\big|\boldsymbol{E}^x_{m,n} - C_m(x)C_n(x)N^2p\big| \geq N^2p(\hat{r}^{-1}\varepsilon - \hat{r}^{-1}\varepsilon\delta - \delta)\Big). && (5.72)
\end{aligned}$$

In order to ensure that $\hat{r}^{-1}\varepsilon - \hat{r}^{-1}\varepsilon\delta - \delta > 0$ for the given $\varepsilon \in (0, \hat{r})$, choose $\delta = \hat{r}^{-1}\varepsilon/2$, i.e., $(\hat{r}^{-1}\varepsilon - \hat{r}^{-1}\varepsilon\delta - \delta) = (\hat{r}^{-1}\varepsilon - \hat{r}^{-2}\varepsilon^2)/2$. Applying the union bound and lemma 5.8 yields

$$\begin{aligned}
&\mathbb{P}\Big(\max_{x\in[M]^N}\max_{m\neq n} r_{m,n}\Big|\frac{1}{N}\sum_{i:x_i=m}\frac{d^{\boldsymbol{G}}_{i,n}(x)}{(1\pm\delta)Np} - C_m(x)C_n(x)\Big| \geq \varepsilon\Big) && (5.73)\\
&\leq 2M^N M(M-1)\exp\Big(-\frac{(N^2p(\hat{r}^{-1}\varepsilon - \hat{r}^{-2}\varepsilon^2)/2))^2}{3N^2p}\Big) && (5.74)\\
&\leq 2M^{N+2}\exp\Big(-\frac{1}{12}N^2p\Big(\frac{\varepsilon}{\hat{r}} - \frac{\varepsilon^2}{\hat{r}^2}\Big)^2\Big). && (5.75)
\end{aligned}$$

Moreover, consider that

$$\begin{aligned}
\mathbb{P}(\mathcal{A}\cap\mathcal{B}) &= \mathbb{P}\Big(\max_{x\in[M]^N}\max_{m\neq n} r_{m,n}\Big|\frac{1}{N}\sum_{i:x_i=m}\frac{d^{\boldsymbol{G}}_{i,n}(x)}{d^{\boldsymbol{G}}_i} - C_m(x)C_n(x)\Big| \geq \varepsilon\\
&\qquad\qquad \text{and } \forall i : (1-\delta)Np \leq d^{\boldsymbol{G}}_i \leq (1+\delta)Np\Big) \qquad (5.76)\\
&\leq \mathbb{P}\Big(\max_{x\in[M]^N}\max_{m\neq n}\max_{\xi_1,\ldots,\xi_N\in[-1,1]} r_{m,n}\Big|\frac{1}{N}\sum_{i:x_i=m}\frac{d^{\boldsymbol{G}}_{i,n}(x)}{(1+\xi_i\delta)Np} - C_m(x)C_n(x)\Big| \geq \varepsilon\Big).
\end{aligned}$$

Since the right term in the absolute value, $C_m(x)C_n(x)$, is independent of i, the maximum is reached by either making the left term, $\sum_{i:x_i=m} d^{\boldsymbol{G}}_{i,n}(x)/(1+\xi_i\delta)Np$, as large as possible or as small as possible, i.e., either all $\xi_i = 1$ or all $\xi_i = -1$. This implies, again using the union bound, that

$$\begin{aligned}
&\mathbb{P}(\mathcal{A}\cap\mathcal{B}) \\
&\leq \mathbb{P}\Big(\max_{\xi\in\{-1,1\}}\max_{x\in[M]^N}\max_{m\neq n} r_{m,n}\Big|\frac{1}{N}\sum_{i:x_i=m}\frac{d_{i,n}^{\boldsymbol{G}}(x)}{(1+\xi\delta)Np} - C_m(x)C_n(x)\Big| \geq \varepsilon\Big) &(5.77)\\
&\overset{(5.75)}{\leq} 4M^{N+2}\exp\Big(-\frac{1}{12}N^2p\Big(\frac{\varepsilon}{\hat{r}}-\frac{\varepsilon^2}{\hat{r}^2}\Big)^2\Big). &(5.78)
\end{aligned}$$

Finally, combing the above bounds yields

$$\begin{aligned}
&\mathbb{P}(\mathcal{A}) \leq \mathbb{P}(\mathcal{A}\cap\mathcal{B}) + \mathbb{P}(\mathcal{B}^C) &(5.79)\\
&\overset{(5.65),(5.78)}{\leq} 4M^{N+2}\exp\Big(-\frac{1}{12}N^2p\Big(\frac{\varepsilon}{\hat{r}}-\frac{\varepsilon^2}{\hat{r}^2}\Big)^2\Big) + 2N\exp\Big(-N\frac{\varepsilon^2 p}{12\hat{r}}+\frac{\varepsilon}{3\hat{r}}\Big), &(5.80)
\end{aligned}$$

which concludes the proof. □

By examining the bounding function f_ε that was derived in the previous proposition, it can be shown for which edge densities $p = p(N)$ the mean-field result holds. The following theorem states that ER random graphs of intermediate density are sufficient to obtain the mean-field limit. Interestingly, the derived threshold for the edge density p is exactly the sharp threshold that yields (asymptotically almost surely) connectedness of $\boldsymbol{G}_{N,p}$ [109].

Theorem 5.11. *Let the edge probability $p = p(N)$ of the ER random graph $\boldsymbol{G}_{N,p}$ be a function of the number of vertices N. If p dominates $\log(N)/N$ asymptotically, i.e.,*

$$p = \omega\Big(\frac{\log N}{N}\Big) \quad \text{as } N\to\infty, \tag{5.81}$$

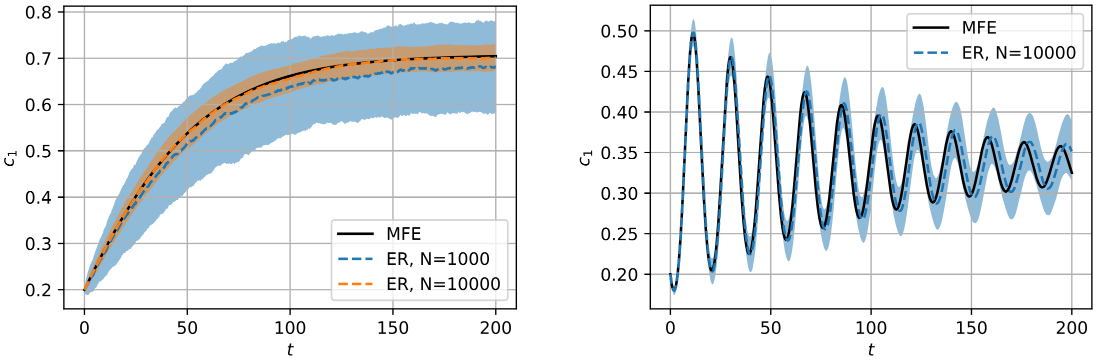
then the dynamics of the opinion shares in the CNVM converges to a mean-field limit as $N\to\infty$, in the sense of both theorem 5.2 (annealed result) and corollary 5.4 (quenched result). The mean-field solution satisfies the ODE

$$\frac{d}{dt}c(t) = \sum_{m\neq n} c_m\big(r_{m,n}c_n + \tilde{r}_{m,n}\big)(e_n - e_m). \tag{5.82}$$

Proof. Recall the bounding function derived in proposition 5.10

$$f_\varepsilon(N) = 4M^{N+2}\exp\Big(-\frac{1}{12}N^2p\Big(\frac{\varepsilon}{\hat{r}}-\frac{\varepsilon^2}{\hat{r}^2}\Big)^2\Big) + 2N\exp\Big(-N\frac{\varepsilon^2 p}{12\hat{r}}+\frac{\varepsilon}{3\hat{r}}\Big). \tag{5.83}$$

In order to apply theorem 5.2, $f_\varepsilon(N)$ has to converge to 0 as $N\to\infty$. For the right-hand term in f_ε to converge to 0, it is necessary that Np dominates $\log N$, which is given for $p=\omega\big(\frac{\log N}{N}\big)$. Similarly, for the left-hand term in f_ε to converge to 0, it is necessary that N^2p dominates N, which is less restrictive and also true for $p=\omega\big(\frac{\log N}{N}\big)$. Moreover, neglecting constants, the right-hand term of f_ε satisfies $N\exp(-Np) \ll N^{-2}$, from which the condition $\sum_N f_\varepsilon(N) < \infty$ of the quenched result (corollary 5.4) follows. □



(a) $M = 2$ opinions

(b) $M = 3$ opinions

Figure 5.2.: Mean (dashed line) ± standard deviation (shaded area) of the CNVM on ER random graphs with edge probability $p = 0.01$, estimated from 200 numerical simulations of the model, in comparison with the mean-field solution (MFE) (5.82). Reprinted from [1].

Example. Numerical results for two example models can be found in figure 5.2, indicating how the derived mean-field solution becomes a good approximation of the stochastic process $\boldsymbol{c}(t)$ for large numbers of agents N. For every sample of the process, a new random graph was generated and then the CNVM was simulated on that graph using the stochastic simulation algorithm presented in chapter 4. The first example (see figure 5.2a) shows the CNVM with $M = 2$ opinions, initial conditions $c(0) = (0.2, 0.8)$, and rate constants

$$r = \begin{pmatrix} - & 0.99 \\ 1 & - \end{pmatrix}, \quad \tilde{r} = \begin{pmatrix} - & 0.01 \\ 0.01 & - \end{pmatrix}. \tag{5.84}$$

In the second example (see figure 5.2b) there are $M = 3$ opinions, the initial condition is $c(0) = (0.2, 0.5, 0.3)$, and rate constants are

$$r = \begin{pmatrix} - & 0.8 & 0.2 \\ 0.2 & - & 0.8 \\ 0.8 & 0.2 & - \end{pmatrix}, \quad \tilde{r}_{m,n} = 0.01 \text{ for all } m \neq n. \tag{5.85}$$

For both examples the edge density was set to $p = 0.01$. Note that if the number of agents is too small, the mean-field equation fails to approximate the dynamics well because of the high variance of $\boldsymbol{c}(t)$, and also the mean of $\boldsymbol{c}(t)$ may not be close to the mean-field solution. As the number of agents increases, the variance of the process decreases and the mean moves closer to the mean-field solution, see figure 5.2a. Moreover, note that the quality of the approximation of $\boldsymbol{c}(t)$ by the mean-field limit may deteriorate over time, as indicated by equation (5.31), which can be seen in figure 5.2b.

Rate of convergence. Investigation of the function f_ε also allows to bound the rate of the convergence to the mean-field limit.

Proposition 5.12. *Consider ER random graphs $\boldsymbol{G}_{N,p}$ with a fixed constant edge probability p. Let the stochastic process $\boldsymbol{c}^N(t)$ denote the CNVM on $\boldsymbol{G}_{N,p}$, and $c(t)$ the solution of the mean-field equation. Then*

$$\mathbb{E}\Big[\sup_{0\le s\le t} \|\boldsymbol{c}^N(s) - c(s)\|\Big] = \mathcal{O}\Big(\frac{\log N}{\sqrt{N}}\Big) \quad \textit{as } N \to \infty. \tag{5.86}$$

Proof. Due to proposition 5.7, it remains to be shown that $f_{\varepsilon(N)}(N) = \mathcal{O}(\varepsilon(N))$ for $\varepsilon(N) := \log N/\sqrt{N}$. Neglecting constants, the bounding function is asymptotically

$$f_\varepsilon(N) \approx \exp(N - N^2(\varepsilon - \varepsilon^2)^2) + N\exp(-N\varepsilon^2). \tag{5.87}$$

Inserting $\varepsilon(N)$ into the right-hand term yields $N\exp(-(\log N)^2)$. Consider that for sufficiently large N

$$\log N + \log\left(\frac{\sqrt{N}}{\log N}\right) \le (\log N)^2 \tag{5.88}$$

$$\implies \quad \log N - (\log N)^2 \le \log\left(\frac{\log N}{\sqrt{N}}\right) \tag{5.89}$$

$$\implies \quad N\exp(-(\log N)^2) \le \frac{\log N}{\sqrt{N}} = \varepsilon(N), \tag{5.90}$$

i.e., the right-hand term is $\mathcal{O}(\varepsilon(N))$. Furthermore, the left-hand term is bounded by

$$\exp(N + 2N^2\varepsilon(N)^3 - N^2\varepsilon(N)^2) = \exp(N + 2\sqrt{N}(\log N)^3 - N(\log N)^2), \tag{5.91}$$

which is $\mathcal{O}(\varepsilon(N))$ due to (for sufficiently large N)

$$N + 2\sqrt{N}(\log N)^3 + \log\left(\frac{\sqrt{N}}{\log N}\right) \le N(\log N)^2 \tag{5.92}$$

$$\implies \quad N + 2\sqrt{N}(\log N)^3 - N(\log N)^2 \le \log\left(\frac{\log N}{\sqrt{N}}\right) \tag{5.93}$$

$$\implies \quad \exp(N + 2\sqrt{N}(\log N)^3 - N(\log N)^2) \le \frac{\log N}{\sqrt{N}} = \varepsilon(N). \tag{5.94}$$

□

It should be noted that, as the above proof suggests, the bound $\log N/\sqrt{N}$ for the rate of convergence is not sharp. It was chosen out of convenience such that the proof is relatively simple. However, as proposition 5.7 shows, a bound for the rate of convergence can never be smaller than $1/\sqrt{N}$ due to the central limit theorem. Thus, the sharp bound is somewhere in between.

These theoretical findings are confirmed by numerical simulations of the model, see figure 5.3. The figure shows that the error between stochastic process and mean-field limit decreases approximately at the rate $1/\sqrt{N}$ for the edge densities $p = 1$ (complete graph) and $p = \log(N)^2/N$. Although the error seems to also decrease at that rate for $p = 10/N$ initially, it stops decreasing for larger N. This initial reduction of the error is merely due to the decrease of variance. But, as the right-hand plot shows, there is a discrepancy between the mean realization for $p = 10/N$ and the mean-field limit that does not seem to vanish as N increases. Hence, the data suggests that in the sparse setting $p = 10/N$ convergence to the MFE can not be observed.

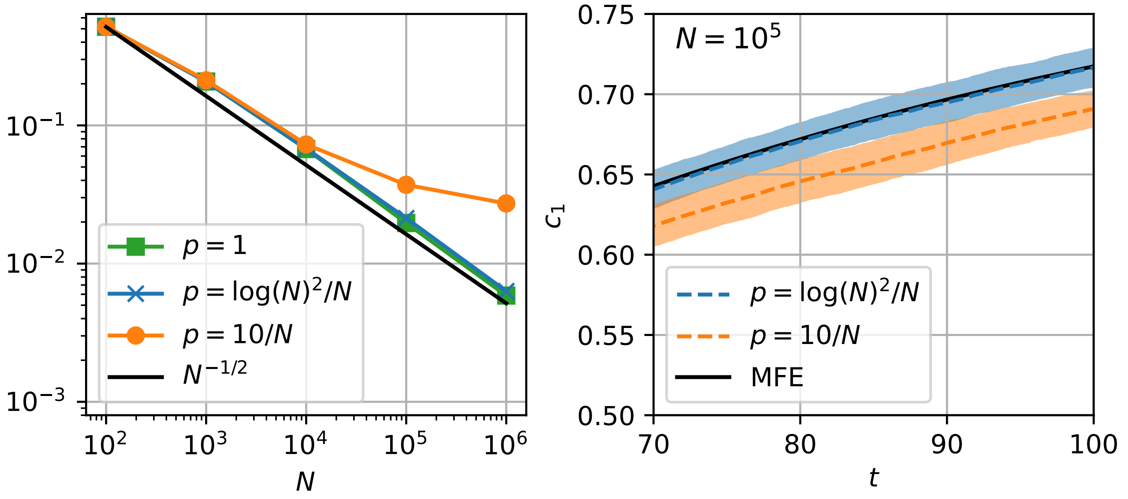


Figure 5.3.: Left: error $\mathbb{E}[\sup_{0\leq s\leq 100}\|\boldsymbol{c}^N(s) - c(s)\|]$ as the number of nodes N increases, for different edge densities p, estimated from numerical model simulations. Right: mean (dashed line) ± standard deviation (shaded area) of the first opinion share c_1 for $N = 10^5$.

5.2.2. Heterogeneous population

Similarly to the previous section, consider a sequence of Erdős–Rényi (ER) random graphs $(\boldsymbol{G}_{N,p})_{N\in\mathbb{N}}$. However, it is now allowed that the population is heterogeneous, i.e., there are K distinct classes of agents that differ by their rate constants $r^k_{m,n}$ and $\tilde{r}^k_{m,n}$, $k = 1,\dots,K$. Thus, for a given graph G an agent i of class k has the rate matrix

$$\left(Q_i^G(x)\right)_{m,n} = r^k_{m,n}\frac{d^G_{i,n}(x)}{d^G_i} + \tilde{r}^k_{m,n}\,, \quad m \neq n. \tag{5.95}$$

As a consequence, the collective variable $C_{(m,k)}(x)$ then measures the share of agents that have opinion m and class k. Note that the quantity of agents in each class and the assignment of agents to the classes can be arbitrary, as long as the initial shares $\boldsymbol{c}^N(0)$ converge to a constant vector c_0 in the large population limit $N \to \infty$ (see the conditions of theorem 5.2). This implies that also the shares of each class k, i.e., the percentages of agents in each class, have to converge in the large population limit. Note however that, as mentioned in the beginning of this chapter, the class assignment is not allowed to depend on the realization of the random graph. In other words, for each N there is a deterministic assignment of the nodes to the classes, while the edges are drawn afterwards and at random.

Again, the mean-field solution will first be derived in a heuristic manner. Consider the propensity functions

$$\alpha^G_{(m,k)\to n}(x) = \sum_{i:(x_i,s_i)=(m,k)} \left(Q_i^G(x)\right)_{m,n} \tag{5.96}$$

$$= \sum_{i:(x_i,s_i)=(m,k)} \left(r^k_{m,n}\frac{d^G_{i,n}(x)}{d^G_i} + \tilde{r}^k_{m,n}\right). \tag{5.97}$$

Because of the homogeneous nature of the ER random graph $\boldsymbol{G}$, the share of agents of opinion n in the neighborhood of agent i, $d^{\boldsymbol{G}}_{i,n}(x)/d^{\boldsymbol{G}}_i$, is expected to be approximately equal to the share of opinion n in the whole system, $c_n := \sum_{k\in[K]} C_{(n,k)}(x)$. Hence, it follows that

$$\frac{1}{N}\alpha^{\boldsymbol{G}}_{(m,k)\to n}(x) \approx \frac{1}{N}\sum_{i:(x_i,s_i)=(m,k)} \Big(r^k_{m,n}c_n + \tilde{r}^k_{m,n}\Big) \tag{5.98}$$

$$= c_{(m,k)}\big(r^k_{m,n}c_n + \tilde{r}^k_{m,n}\big) =: \tilde{\alpha}_{(m,k)\to n}(c) \tag{5.99}$$

which yields the following mean-field ODE when inserted into (5.11):

$$\frac{\mathrm{d}}{\mathrm{d}t}c(t) = \sum_{(m,k)\to n} c_{(m,k)}(t)\Big(r^k_{m,n}\sum_{k'\in[K]} c_{(n,k')}(t) + \tilde{r}^k_{m,n}\Big)\big(e_{(n,k)} - e_{(m,k)}\big). \tag{5.100}$$

Theorem 5.13. *Consider the heterogeneous CNVM as introduced above on the sequence of Erdős–Rényi random graphs $(\boldsymbol{G}_{N,p})_{N\in\mathbb{N}}$. Let the edge probability $p = p(N)$ be a function of the number of vertices N. If p dominates $\log(N)/N$ asymptotically, i.e.,*

$$p = \omega\Big(\frac{\log N}{N}\Big) \quad \text{as } N \to \infty, \tag{5.101}$$

then the dynamics of the collective variables $c = C(x)$, where $c_{(m,k)}$ denotes the share of agents that have opinion m and class k, converges to a mean-field limit as $N \to \infty$, in the sense of both theorem 5.2 (annealed result) and corollary 5.4 (quenched result). The mean-field solution satisfies the ODE (5.100).

Proof. The proof is analogous to the proof for a homogeneous population in section 3.1.1. First define

$$\boldsymbol{E}^x_{(m,k)\to n} := \sum_{i:(x_i,s_i)=(m,k)} d^{\boldsymbol{G}_{N,p}}_{i,n}(x) \tag{5.102}$$

as the number of edges between nodes of extended state (m,k) and nodes of opinion n. Then, analogously to lemma 5.8, it can be shown using the Chernoff bound (lemma A.1) that

$$\mathbb{P}\Big(\Big|\boldsymbol{E}^x_{(m,k)\to n} - C_{(m,k)}(x)C_n(x)N^2p\Big| \geq \varepsilon\Big) \leq 2\exp\Big(-\frac{\varepsilon^2}{3N^2p}\Big), \tag{5.103}$$

where $C_n(x) := \sum_{k\in[K]} C_{(n,k)}(x)$. By inserting the propensity functions into (5.9), it follows that

$$\Delta^{\boldsymbol{G}}(x) = \max_{(m,k)\to n} r^k_{m,n}\Big|\frac{1}{N}\sum_{i:(x_i,s_i)=(m,k)} \frac{d^{\boldsymbol{G}}_{i,n}(x)}{d^{\boldsymbol{G}}_i} - C_{(m,k)}(x)C_n(x)\Big|. \tag{5.104}$$

Analogously to the proof of proposition 5.10, define the events $\mathcal{A}$ and $\mathcal{B}$. Then it can be shown by means of (5.103) that

$$\mathbb{P}(\mathcal{A}) \leq \mathbb{P}(\mathcal{A}\cap\mathcal{B}) + \mathbb{P}(\mathcal{B}^C) \tag{5.105}$$

$$\leq f_\varepsilon(N) := 4KM^{N+2}\exp\Big(-\frac{1}{12}N^2p\Big(\frac{\varepsilon}{\hat{r}} - \frac{\varepsilon^2}{\hat{r}^2}\Big)^2\Big) + 2N\exp\Big(-N\frac{\varepsilon^2 p}{12\hat{r}} + \frac{\varepsilon}{3\hat{r}}\Big).$$

The bounding function f_ε is identical to the homogeneous case (5.61) except for the additional factor K, due to the additional maximum over the classes before applying the union bound. □

Numerical results for an example model are shown in figure 5.4.

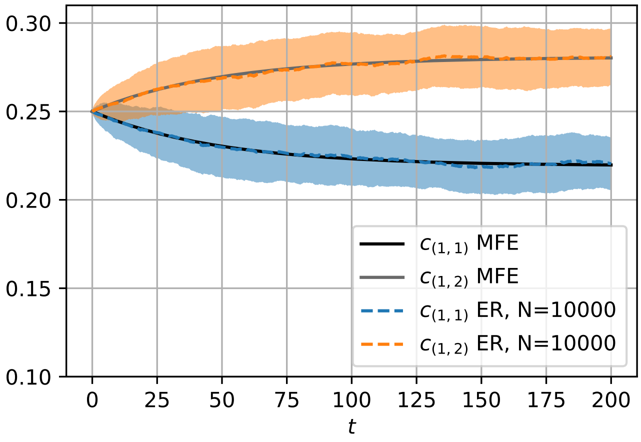


Figure 5.4.: Mean (dashed line) ± standard deviation (shaded area) of the CNVM ($M = 2$ opinions) with heterogeneous population on ER random graphs with $p = 0.01$, estimated from 200 numerical simulations of the model, in comparison with the mean-field solution (5.100). The population consists of $K = 2$ different classes, and the rates r^k are such that class 1 slightly prefers opinion 2 and class 2 prefers opinion 1, i.e., $r^1_{1,2} > r^1_{2,1}$ and $r^2_{1,2} < r^2_{2,1}$. Initial conditions are $(c_{(1,1)}, c_{(1,2)}, c_{(2,1)}, c_{(2,2)}) = (0.25, 0.25, 0.25, 0.25)$. Reprinted from [1].

5.2.3. Stochastic block model

In this section the CNVM (with homogeneous population) is discussed on random graphs given by the stochastic block model. As mentioned in section 3.1.2, in the stochastic block model the population of agents is split into several clusters (blocks) and there are different edge probabilities for connections inside the clusters and for connections between clusters; see figure 5.1 for an example with two clusters. More precisely, let there be K blocks with sizes $b_1, \dots, b_K \in (0, 1] \cap \mathbb{Q}$, such that $\sum_k b_k = 1$. The graphs will be defined on $N_\ell = \ell \cdot \mathrm{LCD}(b_1, \dots, b_k)$ nodes, where LCD refers to the lowest common denominator, such that nodes $\{1, \dots, N_\ell b_1\}$ belong to block 1, nodes $\{N_\ell b_1 + 1, \dots, N_\ell(b_1 + b_2)\}$ to block 2, and so on. Furthermore, let a symmetric matrix of probabilities $(p_{k,k'})_{k,k'=1}^K$ be given, such that $p_{k,k'} \geq 0$ is the probability of an edge between a node in block k and a node in block k'. The edges are then drawn randomly and independently according to these probabilities. It is assumed that for all $k \in [K]$ there is at least one $k' \in [K]$ such that $p_{k,k'} > 0$.

The class of node i is defined as $s_i = k$ if node i is located in the k-th block. Hence, the collective variable $C_{(m,k)}(x)$ measures the share of agents that are located in cluster k and have opinion m.

Heuristic derivation of the mean-field equation. Again, the mean-field solution will first be derived in a heuristic manner. Consider for a given graph G the propensity functions

$$\alpha^G_{(m,k)\to n}(x) = \sum_{i:(x_i,s_i)=(m,k)} \Big(r_{m,n} \frac{d^G_{i,n}(x)}{d^G_i} + \tilde{r}_{m,n} \Big). \tag{5.106}$$

(Note that the population is homogeneous, i.e., every node has the same rate constants r and $\tilde{r}$. It would be straightforward to extend this to the case where nodes have different rate constants r^k and $\tilde{r}^k$ depending on the class k, similarly to section 5.2.2.) For a random graph $\boldsymbol{G}$ generated by the stochastic block model, the degree $d_i^{\boldsymbol{G}}$ of node i in block k is expected to be concentrated around

$$d_i^{\boldsymbol{G}} \approx N_\ell \sum_{k' \in [K]} b_{k'} p_{k,k'} =: N_\ell \bar{p}_k \,. \tag{5.107}$$

Thus, the propensity functions can be approximated by

$$\alpha^{\boldsymbol{G}}_{(m,k)\to n}(x) \approx \frac{r_{m,n}}{N_\ell \bar{p}_k} \sum_{i:(x_i,s_i)=(m,k)} d^{\boldsymbol{G}}_{i,n}(x) + \sum_{i:(x_i,s_i)=(m,k)} \tilde{r}_{m,n} \,. \tag{5.108}$$

Furthermore, the random variable $\sum_{i:(x_i,s_i)=(m,k)} d^{\boldsymbol{G}}_{i,n}(x)$, which counts the number of edges between nodes in cluster k that have opinion m and nodes anywhere that have opinion n, is expected to concentrate around its mean

$$\sum_{i:(x_i,s_i)=(m,k)} d^{\boldsymbol{G}}_{i,n}(x) \approx \sum_{k' \in [K]} c_{(m,k)} c_{(n,k')} N_\ell^2 p_{k,k'}, \tag{5.109}$$

where $c_{(m,k)} \coloneqq C_{(m,k)}(x)$. Hence, it follows that

$$\frac{1}{N_\ell} \alpha^{\boldsymbol{G}}_{(m,k)\to n}(x) \approx c_{(m,k)} \Big(r_{m,n} \frac{\sum_{k' \in [K]} c_{(n,k')} p_{k,k'}}{\bar{p}_k} + \tilde{r}_{m,n} \Big) =: \tilde{\alpha}_{(m,k)\to n}(c), \tag{5.110}$$

from which the mean-field ODE can be obtained by inserting into (5.11):

$$\frac{\mathrm{d}}{\mathrm{d}t} c(t) = \sum_{(m,k)\to n} c_{(m,k)}(t) \Big(r_{m,n} \frac{\sum_{k' \in [K]} c_{(n,k')}(t)\; p_{k,k'}}{\bar{p}_k} + \tilde{r}_{m,n} \Big) \big(e_{(n,k)} - e_{(m,k)} \big), \tag{5.111}$$

where $\bar{p}_k \coloneqq \sum_{k' \in [K]} b_{k'} p_{k,k'}$.

Auxiliary concentration results. Analogously to section 5.2.1, the following auxiliary results are useful to show that the above propensity functions fulfill the conditions of theorem 5.2.

Lemma 5.14. *Given a fixed state $x \in [M]^N$ and the stochastic block model random graph $\boldsymbol{G} = \boldsymbol{G}_{N_\ell}$, define*

$$\boldsymbol{E}^x_{(m,k)\to n} \coloneqq \sum_{i:(x_i,s_i)=(m,k)} d^{\boldsymbol{G}}_{i,n}(x) \tag{5.112}$$

as the number of edges between nodes of extended state (m,k) and nodes of opinion n. Then it follows that

$$\mathbb{E}[\boldsymbol{E}^x_{(m,k)\to n}] = \sum_{k' \in [K]} C_{(m,k)}(x) C_{(n,k')}(x) N^2 p_{k,k'} =: \mu. \tag{5.113}$$

Furthermore, $\boldsymbol{E}^x_{(m,k)\to n}$ satisfies the concentration inequality

$$\mathbb{P}\Big(\Big| \boldsymbol{E}^x_{(m,k)\to n} - \mu \Big| \geq \varepsilon \Big) \leq 2 \exp\Big(- \frac{\varepsilon^2}{3N^2 \bar{p}_k} \Big), \tag{5.114}$$

where again $\bar{p}_k \coloneqq \sum_{k' \in [K]} b_{k'} p_{k,k'}$.

Proof. The number of edges between a node with extended state (m,k) and a node with extended state (n,k') is binomial distributed with $C_{(m,k)}(x)C_{(n,k')}(x)N^2$ trials and success probability $p_{k,k'}$, i.e.,

$$\boldsymbol{E}^x_{(m,k)\to n} \sim \sum_{k'\in[K]} \mathrm{Bin}\big(C_{(m,k)}(x)C_{(n,k')}(x)N^2, p_{k,k'}\big). \tag{5.115}$$

From the Chernoff bound (see lemma A.1) it follows that

$$\mathbb{P}\Big(\Big|\boldsymbol{E}^x_{(m,k)\to n} - \mu\Big| \geq \varepsilon\Big) \leq 2\exp\Big(-\frac{\varepsilon^2}{3\mu}\Big) \leq 2\exp\Big(-\frac{\varepsilon^2}{3N^2\bar{p}_k}\Big), \tag{5.116}$$

where the last inequality is due to $C_{(m,k)}(x)C_{(n,k')}(x) \leq b_{k'}$. □

Moreover, the node degrees are also concentrated in the stochastic block model as the following lemma shows.

Lemma 5.15. *Let node $i \in [N]$ be in cluster k and let $\boldsymbol{d}_i := d_i^{\boldsymbol{G}}$ denote the degree of node i in the stochastic block model. Then for all $\varepsilon > 0$ it holds that*

$$\mathbb{P}\Big(|\boldsymbol{d}_i - N_\ell\bar{p}_k)| \geq \varepsilon N_\ell\bar{p}_k\Big) \leq 2\exp\Big(-N_\ell\frac{\varepsilon^2\bar{p}}{3} + \frac{2\varepsilon}{3}\Big), \tag{5.117}$$

where $\bar{p} := \min_{k\in[K]}\bar{p}_k$.

Proof. Note that $\boldsymbol{d_i}$ is the sum of independent binomial random variables

$$\boldsymbol{d}_i \sim \sum_{k'\in[K]\setminus\{k\}} \mathrm{Bin}(N_\ell b_{k'}, p_{k,k'}) + \mathrm{Bin}(N_\ell b_k - 1, p_{k,k}). \tag{5.118}$$

Using the abbreviation $\mu := \mathbb{E}[\boldsymbol{d}_i]$, it follows that $N_\ell\bar{p}_k = \mu + p_{k,k}$ and

$$\begin{aligned}
\mathbb{P}\Big(|\boldsymbol{d}_i - N_\ell\bar{p}_k)| \geq \varepsilon N_\ell\bar{p}_k\Big) &\leq \mathbb{P}\Big(|\boldsymbol{d}_i - \mu)| + p_{k,k} \geq \varepsilon N_\ell\bar{p}_k\Big) && (5.119)\\
&\leq 2\exp\Big(-\frac{(\varepsilon N_\ell\bar{p}_k - p_{k,k})^2}{3\mu}\Big) && (5.120)\\
&\leq 2\exp\Big(-\frac{(\varepsilon N_\ell\bar{p}_k - p_{k,k})^2}{3N_\ell\bar{p}_k}\Big) && (5.121)\\
&\leq 2\exp\Big(-N_\ell\frac{\varepsilon^2\bar{p}_k}{3} + \frac{2\varepsilon p_{k,k}}{3}\Big) && (5.122)\\
&\leq 2\exp\Big(-N_\ell\frac{\varepsilon^2\bar{p}}{3} + \frac{2\varepsilon}{3}\Big) && (5.123)
\end{aligned}$$

where the second inequality is due to the Chernoff bound (see lemma A.1). □

Large population limit. Using the above auxiliary results the conditions of theorem 5.2 are verified in the following proposition.

Proposition 5.16. *Let $\boldsymbol{G}_\ell$ denote the stochastic block model random graph on N_ℓ nodes, as introduced earlier, and let $\hat r := \max_{m\neq n} r_{m,n}$ and $\bar p := \min_{k\in[K]} \bar p_k$. Then for all $\varepsilon \in (0,\hat r)$ it holds that*

$$\forall \ell\in\mathbb{N}:\ \mathbb{P}\Big(\max_{x\in[M]^{N_\ell}} \Delta^{\boldsymbol{G}_{N_\ell}}(x)\geq \varepsilon\Big)\leq f_\varepsilon(\ell), \tag{5.124}$$

where

$$f_\varepsilon(\ell) := 4M^{N_\ell+2}K\exp\Big(-N_\ell^2\bar p\Big(\frac{\varepsilon}{\hat r}-\frac{\varepsilon^2}{\hat r^2}\Big)^2/12\Big)+2N_\ell\exp\Big(-N_\ell\frac{\varepsilon^2\bar p}{12\hat r}+\frac{\varepsilon}{3\hat r}\Big). \tag{5.125}$$

Proof. Fix any $\ell\in\mathbb{N}$ and denote $\boldsymbol{G} := \boldsymbol{G}_\ell$. Inserting the propensity functions for the stochastic block model given in (5.106) and (5.110) yields

$$\Delta^{\boldsymbol{G}}(x) = \max_{(m,k)\to n} r_{m,n}\Big|\frac{1}{N_\ell}\sum_{i:(x_i,s_i)=(m,k)}\frac{d_{i,n}^{\boldsymbol{G}}}{d_i^{\boldsymbol{G}}} - C_{(m,k)}(x)\frac{\sum_{k'\in[K]}C_{(n,k')}(x)p_{k,k'}}{\bar p_k}\Big|. \tag{5.126}$$

Let $\delta\in(0,1)$ and define the events

$$\mathcal{A} := \Big\{\max_{x\in[M]^N}\Delta^{\boldsymbol{G}}(x)\geq\varepsilon\Big\} \tag{5.127}$$

$$\mathcal{B} := \Big\{\forall i: (1-\delta)N_\ell\bar p_{s_i}\leq d_i^{\boldsymbol{G}}\leq(1+\delta)N_\ell\bar p_{s_i}\Big\}. \tag{5.128}$$

From lemma 5.15 and the union bound it follows that

$$\mathbb{P}(\mathcal{B}^C)\leq 2N_\ell\exp\Big(-\frac{\delta^2 N_\ell\bar p}{3}+\frac{2\delta}{3}\Big), \tag{5.129}$$

where $\mathcal{B}^C$ denotes the complement of $\mathcal{B}$. The goal is now to derive a bound for $\mathbb{P}(\mathcal{A})$ (from which the proposition follows) by showing that $\mathbb{P}(\mathcal{A}\cap\mathcal{B})$ is small and combining that with (5.129). Let $\boldsymbol{E}^x_{(m,k)\to n}$ be as defined in lemma 5.14. For any fixed state $x\in[M]^N$ and any transition $(m,k)\to n$, it follows, using the abbreviation $c_{(m,k)} := C_{(m,k)}(x)$, that

$$\mathbb{P}\Big(r_{m,n}\Big|\frac{1}{N_\ell}\sum_{i:s_i=(m,k)}\frac{d_{i,n}^{\boldsymbol{G}}(x)}{(1\pm\delta)N_\ell\bar p_k} - c_{(m,k)}\frac{\sum_{k'\in[K]}c_{(n,k')}p_{k,k'}}{\bar p_k}\Big|\geq\varepsilon\Big) \tag{5.130}$$

$$= \mathbb{P}\Big(\frac{r_{m,n}}{(1\pm\delta)N_\ell^2\bar p_k}\Big|\boldsymbol{E}^x_{(m,k)\to n} - c_{(m,k)}\sum_{k'\in[K]}c_{(n,k')}p_{k,k'}(1\pm\delta)N_\ell^2\Big|\geq\varepsilon\Big) \tag{5.131}$$

$$\begin{aligned}\leq \mathbb{P}\Big(\Big|\boldsymbol{E}^x_{(m,k)\to n} - c_{(m,k)}\sum_{k'\in[K]}c_{(n,k')}p_{k,k'}N_\ell^2\Big| + \underbrace{c_{(m,k)}\sum_{k'\in[K]}c_{(n,k')}p_{k,k'}}_{\leq\bar p_k}\delta N_\ell^2 \\ \geq r_{m,n}^{-1}(1\pm\delta)N_\ell^2\bar p_k\varepsilon\Big)\end{aligned} \tag{5.132}$$

$$\leq \mathbb{P}\Big(\Big|\boldsymbol{E}^x_{(m,k)\to n} - c_{(m,k)}\sum_{k'\in[K]}c_{(n,k')}p_{k,k'}N_\ell^2\Big|\geq r_{m,n}^{-1}(1\pm\delta)N_\ell^2\bar p_k\varepsilon - \delta N_\ell^2\bar p_k\Big) \tag{5.133}$$

$$\leq \mathbb{P}\Big(\Big|\boldsymbol{E}^x_{(m,k)\to n} - c_{(m,k)}\sum_{k'\in[K]}c_{(n,k')}p_{k,k'}N_\ell^2\Big|\geq N_\ell^2\bar p(\hat r^{-1}\varepsilon - \hat r^{-1}\varepsilon\delta - \delta)\Big). \tag{5.134}$$

Thus, choosing $\delta = \hat{r}^{-1}\varepsilon/2$, as was done in the proof of proposition 5.10 for ER graphs, adding the maxima, and applying lemma 5.14 and the union bound yields

$$\begin{aligned}\mathbb{P}\Big(\max_{x\in[M]^N}\max_{(m,k)\to n} r_{m,n}\Big|\frac{1}{N}\sum_{i:s_i=(m,k)}\frac{d^{\boldsymbol{G}}_{i,n}(x)}{(1\pm\delta)N_\ell\bar{p}_k}\\ - C_{(m,k)}(x)\frac{\sum_{k'\in[K]}C_{(n,k')}(x)p_{k,k'}}{\bar{p}_k}\Big|\geq\varepsilon\Big)\\ \leq 2M^{N_\ell+2}K\exp\Big(-\frac{1}{12}N_\ell^2\bar{p}\Big(\frac{\varepsilon}{\hat{r}}-\frac{\varepsilon^2}{\hat{r}^2}\Big)^2\Big). \qquad (5.135)\end{aligned}$$

With the same arguments as in the proof of proposition 5.10, this leads to

$$\begin{aligned}\mathbb{P}(\mathcal{A}) &\leq \mathbb{P}(\mathcal{A}\cap\mathcal{B})+\mathbb{P}(\mathcal{B}^C) && (5.136)\\ &\leq 4M^{N_\ell+2}K\exp\Big(-\frac{1}{12}N_\ell^2\bar{p}\Big(\frac{\varepsilon}{\hat{r}}-\frac{\varepsilon^2}{\hat{r}^2}\Big)^2\Big)+2N_\ell\exp\Big(-N_\ell\frac{\varepsilon^2\bar{p}}{12\hat{r}}+\frac{\varepsilon}{3\hat{r}}\Big), && (5.137)\end{aligned}$$

which concludes the proof. □

To address the question of how small the edge probabilities $p_{k,k'}$ may become as $\ell\to\infty$ so that convergence to the mean-field limit is guaranteed, it is first assumed that all edge probabilities scale with the same factor. The case that each $p_{k,k'}$ may decrease at its own rate is discussed later.

Theorem 5.17. *Consider the CNVM on a sequence of stochastic block model random graphs $(\boldsymbol{G}_\ell)_{\ell\in\mathbb{N}}$ as defined above. The random graph $\boldsymbol{G}_\ell$ is generated using the scaled edge probabilities $\kappa_\ell\, p_{k,k'}$, where $\kappa_\ell\in[0,1]$ is a scaling factor and $p_{k,k'}$ the fixed edge probability between clusters k and k'. If κ_ℓ dominates $\log(N_\ell)/N_\ell$ asymptotically, i.e.,*

$$\kappa_\ell=\omega\Big(\frac{\log N_\ell}{N_\ell}\Big)\quad \textit{as } \ell\to\infty, \qquad (5.138)$$

then the dynamics of the collective variables $c=C(x)$, where $c_{(m,k)}$ measures the share of agents that have opinion m and are located in the k-th block, converges to a mean-field limit as $\ell\to\infty$, in the sense of both theorem 5.2 (annealed result) and corollary 5.4 (quenched result). The mean-field solution satisfies the ODE

$$\frac{d}{dt}c(t)=\sum_{(m,k)\to n}c_{(m,k)}(t)\Big(r_{m,n}\frac{\sum_{k'\in[K]}c_{(n,k')}(t)\;p_{k,k'}}{\bar{p}_k}+\tilde{r}_{m,n}\Big)\big(e_{(n,k)}-e_{(m,k)}\big), \qquad (5.139)$$

which was derived earlier in equation (5.111).

Proof. The derived bounding function

$$f_\varepsilon(\ell)=4M^{N_\ell+2}K\exp\Big(-\frac{1}{12}N_\ell^2\bar{p}^\ell\Big(\frac{\varepsilon}{\hat{r}}-\frac{\varepsilon^2}{\hat{r}^2}\Big)^2\Big)+2N_\ell\exp\Big(-N_\ell\frac{\varepsilon^2\bar{p}^\ell}{12\hat{r}}+\frac{\varepsilon}{3\hat{r}}\Big), \qquad (5.140)$$

where $\bar{p}^\ell := \kappa_\ell\min_{k\in[K]}\bar{p}_k$, is identical to the bounding function for ER random graphs (see proposition 5.10), except for the additional factor K and the value $\bar{p}^\ell$ instead of p. Hence, the proof is analogous. □

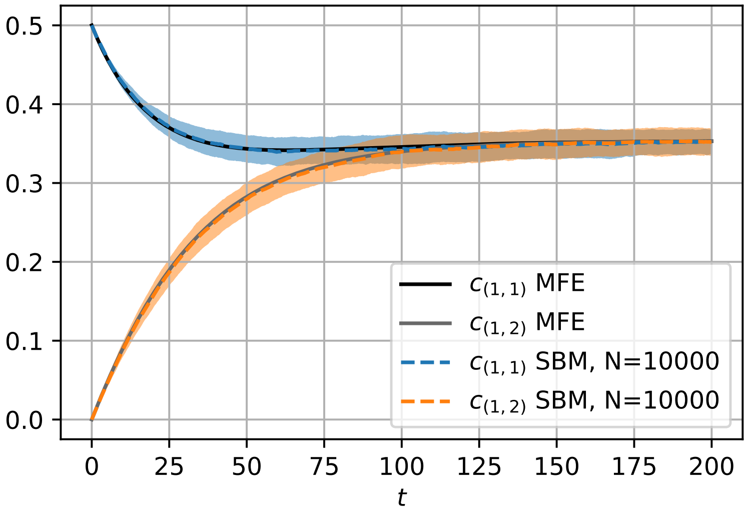


Figure 5.5.: Mean (dotted line) ± standard deviation (shaded area) of the CNVM on a stochastic block model (SBM) with two equal size blocks and $p_{1,1} = p_{2,2} = 0.01$ and $p_{1,2} = 0.0001$, estimated from numerical simulations, in comparison with the mean-field solution (5.139). $M = 2$ opinions with initial conditions $(c_{(1,1)}, c_{(1,2)}, c_{(2,1)}, c_{(2,2)}) = (0.5, 0, 0, 0.5)$. Reprinted from [1].

In the previous theorem the case that all edge probabilities $p_{k,k'}$ are scaled using the same factor κ_ℓ was considered. It is also possible to let each edge probability scale independently. Let $p^\ell_{k,k'}$ define the edge probabilities used to construct the graph $\boldsymbol{G}_\ell$. For the bounding function $f_\varepsilon(\ell)$ in (5.140) to converge to 0 it is then required that

$$\bar{p}^\ell := \min_{k\in[K]} \bar{p}^\ell_k := \min_{k\in[K]} \sum_{k'\in[K]} b_{k'} p^\ell_{k,k'} = \omega\left(\frac{\log N_\ell}{N_\ell}\right), \tag{5.141}$$

which yields the following condition on the $p^\ell_{k,k'}$ for convergence to a mean-field limit:

$$\forall k \in [K]\ \exists k' \in [K] : \quad p^\ell_{k,k'} = \omega\left(\frac{\log N_\ell}{N_\ell}\right) \text{ as } \ell \to \infty. \tag{5.142}$$

Moreover, the mean-field equation (5.139) has to be adapted to this setting: the factor $p_{k,k'}/\bar{p}_k$ in (5.139) has to be replaced by the limit $\lim_{\ell\to\infty} p^\ell_{k,k'}/\bar{p}^\ell_k$ and hence the edge probabilities $p^\ell_{k,k'}$ may only be chosen in such a way that these limits exist for all k, k'. All in all, this means that for the mean-field limit to hold it is sufficient that every cluster is well-connected to at least one other cluster or itself. If a cluster k is only sparsely connected to another cluster k' (or itself), the two are not coupled in the MFE as the factor $p^\ell_{k,k'}/\bar{p}^\ell_k$ vanishes in the limit $\ell \to \infty$.

Numerical results for an example stochastic block model are shown in figure 5.5. In the example, there are two equal size blocks and $M = 2$ opinions. Initially, every agent in block 1 has opinion 1 and every agent in block 2 has opinion 2. Over time the concentrations in both blocks equilibrate.

5.2.4. Random regular graphs

In this section mean-field results for the CNVM on uniformly random regular graphs are derived. A simple graph is called d-regular if every node has exactly the degree d. Denote by $\boldsymbol{G}_{N,d}$ the uniformly random d-regular graph on N nodes, i.e., every d-regular graph has equal probability and every other graph has probability 0. It is again implicitly assumed that the degree $d = d(N)$ may depend on the size of the graph N. Similarly to Erdős–Rényi random graphs, uniformly random d-regular graphs likely have a homogeneous edge density, which indicates a mean-field limit with respect to the simple shares without any further distinction into classes ($K = 1$). Thus, the same propensity functions as in section 5.2.1 will be employed, resulting in the same MFE

$$\frac{\mathrm{d}}{\mathrm{d}t}c(t) = \sum_{m \neq n} c_m(t)\big(r_{m,n}c_n(t) + \tilde{r}_{m,n}\big)(e_n - e_m). \tag{5.143}$$

However, due to the stochastic dependence of edges in $\boldsymbol{G}_{N,d}$ (in contrast to ER random graphs), working with random d-regular graphs is more intricate, especially in the case of small d. In the case of large degrees d on the other hand, the distributions of the random regular graph and the ER random graph with $p = d/N$ become asymptotically identical, which is the subject of the *sandwich conjecture* (see conjecture 3.4). Given that the degrees are large enough such that the sandwich conjecture applies, e.g., $d \approx N^a$ for any fixed $a > 0$, the convergence to the mean-field limit is hence automatically obtained by our previous theorem 5.11 for ER random graphs.

As stated before, the case of small degree d is substantially more difficult. It is easier to deal with the configuration model instead, which was discussed in detail in section 3.1.6, and then transfer the results back to the regular simple graph setting. Recall that in the configuration model d half-edges are assigned to each node. Then repeatedly two half-edges are picked uniformly at random from all remaining half-edges, joined together to form an edge between the nodes they were assigned to, and removed from the set of available half-edges, until all half-edges are used up. This procedure yields a multigraph. A configuration F is defined as a partition of the set of half-edges W into pairs, and a configuration F induces a multigraph $\gamma(F)$. Moreover, every configuration F can be constructed from a tuple $t \in \Pi = \{(t_1, \dots, t_\eta) \mid t_r \in [Nd - 2r + 1] \text{ for all } r\}$, which is denoted by $F = \psi(t)$. Finally, recall that the random variable $\boldsymbol{t}$ that is uniform over Π induces multigraphs $\gamma(\psi(\boldsymbol{t}))$ with the same distribution as the configuration model.

Similarly to the auxiliary concentration results in the previous sections, the following lemma addresses the probability that the number of edges between nodes of different opinions deviates from its expectation.

Lemma 5.18. *Let $x \in [M]^N$ and fix two distinct opinions $m, n \in [M]$. Assume that the state x is ordered, such that the m-opinion nodes are first, the n-opinion nodes come after that, and then the rest. Define the random variable $g(\boldsymbol{t})$ as the number of edges between nodes of opinion m and nodes of opinion n in the induced multigraph $\gamma(\psi(\boldsymbol{t}))$, with respect to x. Then it follows that*

$$\mathbb{P}\left(\left|g(\boldsymbol{t}) - \frac{c_m c_n N^2 d^2}{Nd - 1}\right| \geq \varepsilon\right) \leq 2\exp\Big(-\frac{\varepsilon^2}{4Nd}\Big), \tag{5.144}$$

where c_m, c_n denote the shares of opinions m, n in the state x.

Proof. Assume that there is at least one node with opinion m and at least one node with opinion n in x; otherwise the lemma is trivially true. Consider two tuples $t, t' \in \Pi$ that only differ in one coordinate l, i.e., $t = (t_1, \dots, t_\eta)$, $t' = (t_1, \dots, t_{l-1}, t'_l, t_{l+1}, \dots, t_\eta)$. Let $b_m \in W$ denote the maximal element such that $x_{\varphi(b_m)} = m$ (recall equation (3.8)), and define b_n analogously. Note that, due to the ordering of x, it is $b_m < b_n$. The values b_m and b_n act as important boundaries because the edges counting towards $g(t)$ have to cross b_m but must not cross b_n. (An edge (s, e) with $s, e \in W$ is defined to be crossing the boundary b if $s \leq b < e$.) Lemma A.3 shows that the number of edges crossing any boundary can vary by at most 2 between t and t'. Hence, it follows that $|g(t) - g(t')| \leq 4$ because there are the two boundaries b_m, b_n to consider for $g(t)$. As the random vector $\boldsymbol{t} = (\boldsymbol{t}_1, \dots, \boldsymbol{t}_\eta)$ has independent components, McDiarmid's inequality [145] can be applied:

$$\mathbb{P}\Big(\Big|g(\boldsymbol{t}) - \mathbb{E}[g(\boldsymbol{t})]\Big| \geq \varepsilon\Big) \leq 2\exp\Big(-\frac{2\varepsilon^2}{16\eta}\Big) = 2\exp\Big(-\frac{\varepsilon^2}{4Nd}\Big). \tag{5.145}$$

Finally, note that there are $c_m Nd$ half-edges attached to nodes of opinion m, and each of these has a $c_n Nd/(Nd-1)$ chance to get matched with a half-edge of a node with opinion n. Hence, it follows that

$$\mathbb{E}[g(\boldsymbol{t})] = c_m Nd \frac{c_n Nd}{Nd - 1}, \tag{5.146}$$

which completes the proof. □

Next, the bounding function f_ε from the conditions of theorem 5.2 is derived. Consider a sequence of uniformly random regular graphs $\boldsymbol{G}_\ell := \boldsymbol{G}_{N_\ell, d_\ell}$, $\ell \in \mathbb{N}$, where $d_\ell := d(N_\ell)$. Note that for a given degree d not all graph sizes N are possible, hence the sequence $(N_\ell)_\ell$ is necessary.

Proposition 5.19. *Assume $d_\ell \ll N_\ell^{1/7}$ and denote $\hat{r} := \max_{m \neq n} r_{m,n}$. Then for all $\varepsilon > 0$ there exists a function $f_\varepsilon : \mathbb{N} \to \mathbb{R}_{\geq 0}$ such that*

$$\forall \ell \in \mathbb{N}: \ \mathbb{P}\Big(\max_{x \in [M]^{N_\ell}} \Delta^{\boldsymbol{G}_\ell}(x) \geq \varepsilon\Big) \leq f_\varepsilon(\ell), \tag{5.147}$$

and

$$f_\varepsilon(\ell) = (2 + o(1)) M^{N_\ell + 2} \exp\Big(d_\ell^2 - N_\ell d_\ell \frac{\varepsilon^2}{4\hat{r}^2} + \frac{\varepsilon}{4\hat{r}}\Big) \quad \text{as } \ell \to \infty. \tag{5.148}$$

Proof. Let $\ell \in \mathbb{N}$ and for simpler notation denote $\boldsymbol{G} := \boldsymbol{G}_\ell$, $N := N_\ell$, $d := d_\ell$. Inserting the propensity functions (see equations (5.49) and (5.51)) yields

$$\Delta^{\boldsymbol{G}}(x) = \max_{m \neq n} r_{m,n} \Big| \frac{1}{N} \sum_{i : x_i = m} \frac{d^{\boldsymbol{G}}_{i,n}(x)}{d^{\boldsymbol{G}}_i} - C_m(x) C_n(x) \Big|. \tag{5.149}$$

Fix two opinions $m \neq n$ and let $x \in [M]^N$ be ordered as in lemma 5.18. As all realizations of $\boldsymbol{G}$ are d-regular, it follows that

$$\frac{1}{N} \sum_{i : x_i = m} \frac{d^{\boldsymbol{G}}_{i,n}(x)}{d^{\boldsymbol{G}}_i} = \frac{1}{Nd} \sum_{i : x_i = m} d^{\boldsymbol{G}}_{i,n}(x) =: \frac{1}{Nd} \boldsymbol{E}_{m,n}\,, \tag{5.150}$$

where $\boldsymbol{E}_{m,n}$ denotes the number of edges between nodes of opinion m and nodes of opinion n in $\boldsymbol{G}$, with respect to the state x. Consider also the number of edges between nodes of opinion m and nodes of opinion n in the configuration model, which is denoted by $g(\boldsymbol{t})$ as in lemma 5.18. These two quantities are related as follows, provided $d < N^{1/7}$ [109, Thm 11.3]:

$$\begin{aligned}\mathbb{P}\Big(\Big|\boldsymbol{E}_{m,n} - \frac{c_m c_n N^2 d^2}{Nd-1}\Big| \geq \varepsilon\Big) &\leq (1+o(1))e^{\lambda(\lambda+1)}\mathbb{P}\Big(\Big|g(\boldsymbol{t}) - \frac{c_m c_n N^2 d^2}{Nd-1}\Big| \geq \varepsilon\Big) && (5.151)\\ &\leq (1+o(1))e^{\lambda(\lambda+1)} 2\exp\Big(-\frac{\varepsilon^2}{4Nd}\Big), && (5.152)\end{aligned}$$

where $\lambda := \frac{d-1}{2}$ and the abbreviation $c_m := C_m(x)$ is used. With the notation $\hat{r} := \max_{m\neq n} r_{m,n}$, it follows that

$$\begin{aligned}&\mathbb{P}\Big(r_{m,n}\Big|\frac{1}{N}\sum_{i:x_i=m}\frac{d^{\boldsymbol{G}}_{i,n}(x)}{d^{\boldsymbol{G}}_i} - c_m c_n\Big| \geq \varepsilon\Big) && (5.153)\\ &\leq \mathbb{P}\Big(\Big|\frac{1}{N}\sum_{i:x_i=m}\frac{d^{\boldsymbol{G}}_{i,n}(x)}{d^{\boldsymbol{G}}_i} - \frac{c_m c_n Nd}{Nd-1}\Big| + \Big|\frac{c_m c_n Nd}{Nd-1} - c_m c_n\Big| \geq \frac{\varepsilon}{\hat{r}}\Big) && (5.154)\\ &= \mathbb{P}\Big(\frac{1}{Nd}\Big|\boldsymbol{E}_{m,n} - \frac{c_m c_n N^2 d^2}{Nd-1}\Big| + \frac{c_m c_n}{Nd-1} \geq \frac{\varepsilon}{\hat{r}}\Big) && (5.155)\\ &= \mathbb{P}\Big(\Big|\boldsymbol{E}_{m,n} - \frac{c_m c_n N^2 d^2}{Nd-1}\Big| \geq Nd(\frac{\varepsilon}{\hat{r}} - \frac{c_m c_n}{Nd-1})\Big) && (5.156)\\ &\overset{(5.152)}{\leq} (1+o(1))e^{\lambda(\lambda+1)} 2\exp\Big(-\frac{\big(Nd(\frac{\varepsilon}{\hat{r}} - \frac{c_m c_n}{Nd-1})\big)^2}{4Nd}\Big) && (5.157)\\ &\leq (1+o(1))e^{\lambda(\lambda+1)} 2\exp\Big(-Nd\frac{\varepsilon^2}{4\hat{r}^2} + \frac{\varepsilon}{4\hat{r}}\Big). && (5.158)\end{aligned}$$

Recall that an ordered state x was assumed. However, due to the indifference of the random regular graph with respect to the specific node numbering, a certain regular graph is just as likely as the same graph but with permuted node labels, i.e., the random regular graph is *invariant under graph isomorphism* as defined in section 3.3. Using this property, it follows from proposition 3.10 that the bound (5.158) also holds for general states x. Finally, applying the union bound yields

$$\begin{aligned}&\mathbb{P}\Big(\max_{x\in[M]^N}\Delta^{\boldsymbol{G}}(x) \geq \varepsilon\Big) && (5.159)\\ &= \mathbb{P}\Big(\max_{x\in[M]^N}\max_{m\neq n} r_{m,n}\Big|\frac{1}{N}\sum_{i:x_i=m}\frac{d^{\boldsymbol{G}}_{i,n}(x)}{d^{\boldsymbol{G}}_i} - C_m(x)C_n(x)\Big| \geq \varepsilon\Big) && (5.160)\\ &\overset{(5.158)}{\leq} (1+o(1))M^N M(M-1)e^{\lambda(\lambda+1)} 2\exp\Big(-Nd\frac{\varepsilon^2}{4\hat{r}^2} + \frac{\varepsilon}{4\hat{r}}\Big) && (5.161)\\ &\leq (2+o(1))M^{N+2}\exp\Big(d^2 - Nd\frac{\varepsilon^2}{4\hat{r}^2} + \frac{\varepsilon}{4\hat{r}}\Big), && (5.162)\end{aligned}$$

which concludes the proof. □

Theorem 5.20. *Consider the CNVM on a sequence of uniformly random regular graphs* $\boldsymbol{G}_\ell = \boldsymbol{G}_{N_\ell, d_\ell}$, $\ell \in \mathbb{N}$, *such that*

$$\lim_{\ell \to \infty} d_\ell = \infty. \tag{5.163}$$

Then the dynamics of the opinion shares in the CNVM converges to a mean-field limit as $\ell \to \infty$, *in the sense of both theorem 5.2 (annealed result) and corollary 5.4 (quenched result). The associated mean-field ODE is*

$$\frac{d}{dt} c(t) = \sum_{m \neq n} c_m(t) \big(r_{m,n} c_n(t) + \tilde{r}_{m,n} \big) (e_n - e_m). \tag{5.164}$$

Proof. In the case of small degrees $d_\ell \ll (N_\ell)^{1/7}$, proposition 5.19 applies. Neglecting constants, the function $f_\varepsilon(\ell) \approx \exp(N_\ell + d_\ell^2 - N_\ell d_\ell)$ from the proposition is asymptotically bounded from above by $(N_\ell)^{-2}$ since $d_\ell \to \infty$. Hence, the conditions from theorem 5.2 and corollary 5.4 are satisfied. In the case of large degrees, e.g., $d_\ell \gg (N_\ell)^{1/8}$, the statement follows from the sandwich conjecture 3.4 as explained at the beginning of this section. □

Recall that in section 5.2.1 the convergence to the MFE for Erdős–Rényi random graphs was shown under the condition that the average node degree grows faster than $\log(N)$, whereas the above result for d-regular graphs only requires unboundedness of the degree d, i.e., it can grow arbitrarily slowly. Intuitively, the regularity of the graphs allows for the mean-field limit to hold under more sparsity.

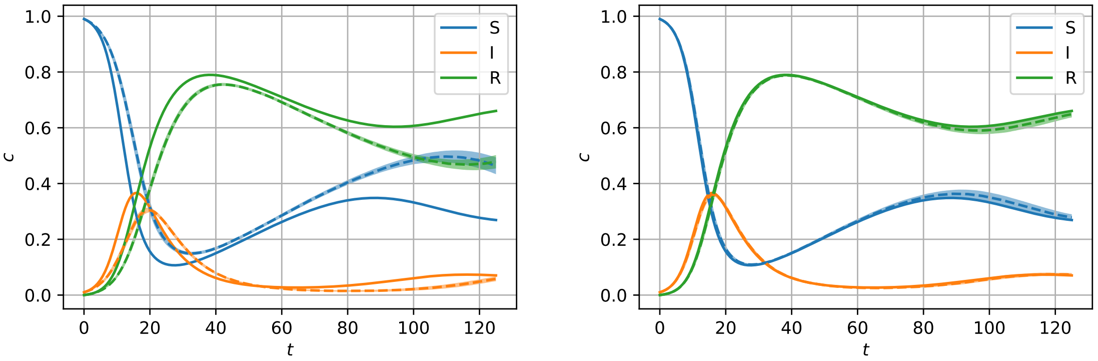
A numerical example for uniformly random regular graphs is shown in figure 5.6. The rates r and $\tilde{r}$ were chosen so that the CNVM produces a so-called SIRS model, i.e., susceptible nodes are infected by infectious neighbors, infectious nodes randomly become recovered, and recovered nodes randomly become susceptible again. Written in the order (S,I,R), the rate matrices are of the form

$$r = \begin{pmatrix} - & r_{S,I} & 0 \\ 0 & - & 0 \\ 0 & 0 & - \end{pmatrix}, \quad \tilde{r} = \begin{pmatrix} - & 0 & 0 \\ 0 & - & \tilde{r}_{I,R} \\ \tilde{r}_{R,S} & 0 & - \end{pmatrix}. \tag{5.165}$$

In the example, a steep wave of infections is observed followed by a smaller second wave. The figure illustrates how the discrepancy between model realizations and mean-field solution decreases when the degree d increases, as indicated by theorem 5.20. For $d = 10$, the approximation quality of the mean-field solution is poor, even though the number of agents $N = 10000$ is quite large. (Increasing N reduces the variance of the model realizations, but does not necessarily move the mean closer to the mean-field solution.) For $d = 100$ on the other hand, the mean-field limit is a reasonable approximation. Hence, in order to achieve a certain approximation quality it is crucial that both N and d are sufficiently large.

5.3. Medium-sized populations

In the previous sections it was shown that, for certain networks, the mean-field equation (MFE) describes the dynamics of the *shares* $c(t)$ in the large population limit. In the



(a) Degree $d = 10$.

(b) Degree $d = 100$.

Figure 5.6.: Mean (dotted line) ± standard deviation (shaded area) of an SIRS model on uniformly random regular graphs with $N = 10000$ nodes, estimated from 100 realizations, in comparison to the mean-field solution (5.164) (solid line). Initial shares $(c_S, c_I, c_R) = (0.99, 0.01, 0)$. Reprinted from [1].

context of medium-sized populations however, an approximation by the MFE is not advisable due to the significant stochastic fluctuations that are not captured by the MFE. It is not even guaranteed that the MFE is close to the mean of the process $\boldsymbol{c}(t)$, especially for smaller populations. The objective of this section is to derive an approximation similar to the MFE that is better-suited for this mesoscopic scale.

For simplicity, assume that there is only one class $K = 1$, i.e., $\boldsymbol{c}(t)$ describes the global shares in the system. Recall the propensity functions

$$\alpha_{m\to n}(x) \coloneqq \sum_{i:x_i=m} \big(Q_i(x)\big)_{m,n} \tag{5.166}$$

and assume again that there exist reduced propensity functions $\tilde{\alpha}_{m\to n}$ such that

$$\frac{1}{N}\alpha_{m\to n}(x) \approx \tilde{\alpha}_{m\to n}(C(x)), \tag{5.167}$$

where $C(x)$ are the shares associated to x. Then the MFE is given by

$$\frac{\mathrm{d}}{\mathrm{d}t}c(t) = \sum_{m\to n} \tilde{\alpha}_{m\to n}(c(t))\, v_{m\to n} =: F(c(t)), \tag{5.168}$$

where $v_{m\to n}$ are the state-change vectors, see section 5.1. Moreover, recall that the projected process $\boldsymbol{c}(t) \coloneqq C(\boldsymbol{x}(t))$ can be written as

$$\boldsymbol{c}(t+\tau) = \boldsymbol{c}(t) + \sum_{m\to n} \boldsymbol{P}_{m\to n}\Big(\int_t^{t+\tau} \alpha_{m\to n}(\boldsymbol{x}(s))\,\mathrm{d}s\Big)\frac{v_{m\to n}}{N}. \tag{5.169}$$

By assuming that $\tau > 0$ is small enough such that the propensity functions stay approximately constant between t and $t+\tau$, this simplifies to

$$\boldsymbol{c}(t+\tau) \approx \boldsymbol{c}(t) + \sum_{m\to n} \boldsymbol{P}_{m\to n}\big(\tau\, \alpha_{m\to n}(\boldsymbol{x}(t))\big)\frac{v_{m\to n}}{N}. \tag{5.170}$$

Note that the random variable $\boldsymbol{P}_{m\to n}\big(\tau\,\alpha_{m\to n}(\boldsymbol{x}(t))\big)$ is Poisson distributed with parameter $\boldsymbol{\lambda} := \tau\,\alpha_{m\to n}(\boldsymbol{x}(t))$. If the parameter $\boldsymbol{\lambda}$ is large enough, the Poisson distribution can be approximated well by a normal distribution $\boldsymbol{N}_{m\to n}(\boldsymbol{\lambda}, \boldsymbol{\lambda})$ with mean $\boldsymbol{\lambda}$ and variance $\boldsymbol{\lambda}$, see for example [146, Appendix A]. As τ is assumed to be small, the propensity $\alpha_{m\to n}(\boldsymbol{x}(t))$ has to be quite large for $\boldsymbol{\lambda}$ to be large enough such that this approximation holds. Hence, the larger the number of agents is, the larger the propensities are, and the better the above approximation works. Inserting into (5.170) yields

$$\boldsymbol{c}(t+\tau) \approx \boldsymbol{c}(t) + \sum_{m\to n} \boldsymbol{N}_{m\to n}\big(\tau\,\alpha_{m\to n}(\boldsymbol{x}(t)), \tau\,\alpha_{m\to n}(\boldsymbol{x}(t))\big)\,\frac{v_{m\to n}}{N} \tag{5.171}$$

and by converting to the standard normal distribution it follows that

$$\begin{aligned}\boldsymbol{c}(t+\tau) \approx \boldsymbol{c}(t) &+ \sum_{m\to n} \tau\,\alpha_{m\to n}(\boldsymbol{x}(t))\,\frac{v_{m\to n}}{N} \\ &+ \sum_{m\to n} \sqrt{\tau\,\alpha_{m\to n}(\boldsymbol{x}(t))}\;\boldsymbol{N}_{m\to n}(0,1)\,\frac{v_{m\to n}}{N}.\end{aligned} \tag{5.172}$$

In the next step, the reduced propensity functions $N\tilde{\alpha}_{m\to n}(\boldsymbol{c}(t)) \approx \alpha_{m\to n}(\boldsymbol{x}(t))$ are inserted, which yields

$$\begin{aligned}\boldsymbol{c}(t+\tau) \approx \boldsymbol{c}(t) &+ \sum_{m\to n} \tau\,\tilde{\alpha}_{m\to n}(\boldsymbol{c}(t))\,v_{m\to n} \\ &+ \sum_{m\to n} \sqrt{\tau\,\tilde{\alpha}_{m\to n}(\boldsymbol{c}(t))}\;\boldsymbol{N}_{m\to n}(0,1)\,\frac{v_{m\to n}}{\sqrt{N}}.\end{aligned} \tag{5.173}$$

Thus, written in the form of a stochastic differential equation, the evolution of the shares $\boldsymbol{c}(t)$ is approximately given by

$$\mathrm{d}\boldsymbol{c}(t) = \underbrace{\sum_{m\to n} \tilde{\alpha}_{m\to n}(\boldsymbol{c}(t))\,v_{m\to n}}_{F(\boldsymbol{c}(t))}\,\mathrm{d}t + \sum_{m\to n} \frac{1}{\sqrt{N}}\sqrt{\tilde{\alpha}_{m\to n}(\boldsymbol{c}(t))}\,v_{m\to n}\,\mathrm{d}\boldsymbol{W}_{m\to n}(t), \tag{5.174}$$

where $\boldsymbol{W}_{m\to n}$ are independent Wiener processes. In the context of chemical reaction systems the above equation (5.174) is known as the *chemical Langevin equation* (CLE) [146, 40]. The CLE extends the mean-field equation given by $F(\boldsymbol{c}(t))$ by adding a stochastic diffusive term that vanishes at the rate $N^{-1/2}$ as $N \to \infty$. This diffusive term not only allows the replication of the stochastic fluctuations for intermediate populations, but also results in a mean of the CLE that is closer to the mean of the actual model than the solution of the MFE.

From the derivation of the CLE presented above it is apparent that the CLE is a valid approximation under similar conditions like the conditions for the MFE, e.g., the existence of reduced propensity functions $N\tilde{\alpha}_{m\to n}(\boldsymbol{c}(t)) \approx \alpha_{m\to n}(\boldsymbol{x}(t))$ is required. Due to the considerations presented previously this tends to be the case for sufficiently homogeneous and dense networks like Erdős–Rényi random graphs or random regular graphs with large enough (expected) node degrees, see section 5.2. A precise investigation of the approximation error of the CLE depending on the underlying network could be the subject of future work.

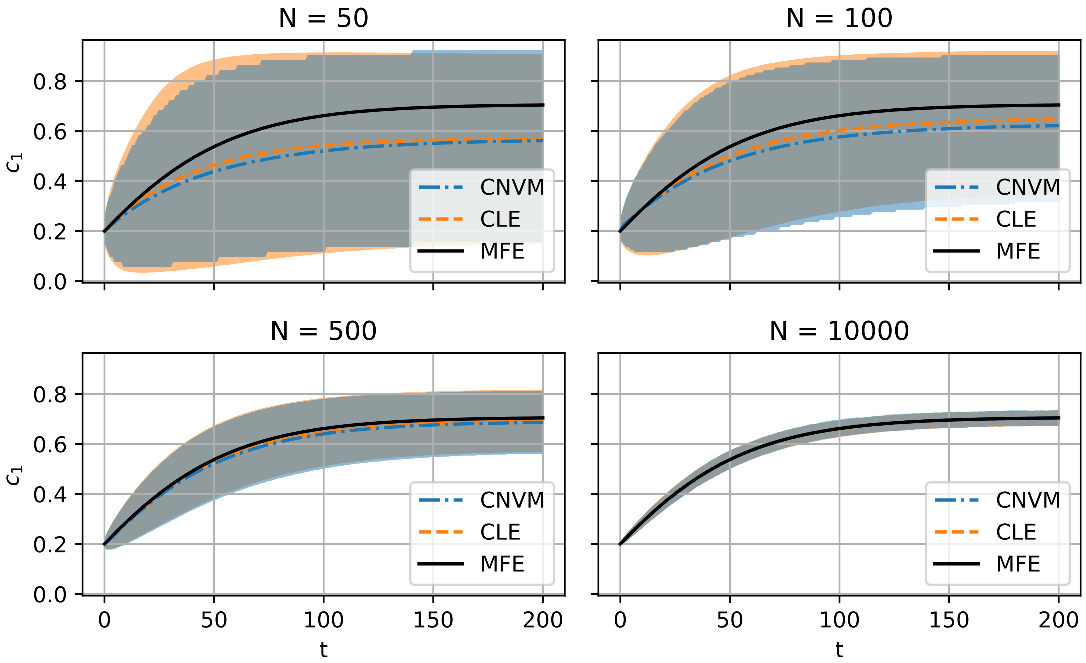


Figure 5.7.: CNVM on Erdős–Rényi random graphs of different sizes N and $p = 0.05$, and approximation with CLE (5.174) and MFE (5.168). The plot shows the mean and a shaded area that contains 2/3 of the probability mass.

Instead, a numerical example is presented in the following. Consider the CNVM with $M = 2$ opinions and parameters

$$r = \begin{pmatrix} - & 1.01 \\ 0.99 & - \end{pmatrix}, \qquad \tilde{r}_{m,n} = 0.01. \tag{5.175}$$

Simulations on Erdős–Rényi random graphs of different sizes and a fixed edge density $p = 0.05$ are compared with the CLE and the MFE in figure 5.7. For $N = 50$ and $N = 100$, the CLE is already a decent approximation as both the mean and the variance match quite closely. Note also that the MFE is a bad estimation for the mean of the process for these small N. This changes at $N \approx 500$, where the mean of the CNVM, the mean of the CLE, and the MFE are all close together. However, the CLE still has the advantage of capturing the variance of the process in contrast to the MFE. Not surprisingly, for large $N > 10^5$ the variance is virtually negligible such that the CLE offers no advantage over the much simpler MFE.

5.4. Hybrid models for leader-follower dynamics

This section explores the scenario of a few distinguished and very influential agents called *leaders* affecting a large mass of ordinary agents called *followers*. Especially since “influencers” became relevant in social media, this theme is studied extensively in the literature, see for example [147, 148, 149]. However, the leaders can also be utilized to model a general sentiment slightly pulling the system into a certain direction, for instance due to external perturbations that are not modeled in the system itself [150]. The considerations presented in this section are motivated by reference [151], where a similar hybrid model is derived for systems describing chemical reactions.

Motivating example: star graph. On a star graph, where node 1 is the central node and all other nodes $i > 1$ are connected only to that central node, it is apparent that the central node has a much larger impact on the macroscopic dynamics than the others. Consider for instance the continuous-time noisy voter model (CNVM) on such a star graph. Recall that in the CNVM the rate at which a node i transitions from one opinion $m \in [M]$ to a different opinion $n \in [M]$ is given by

$$r_{m,n} \frac{d_{i,n}(x)}{d_i} + \tilde{r}_{m,n}, \tag{5.176}$$

see section 2.2 for details. For the star graph it is easy to see that

$$d_i = \begin{cases} 1, & i \neq 1 \\ N-1, & i = 1 \end{cases} \quad \text{and} \quad d_{i,n}(x) = \begin{cases} 1, & i \neq 1, x_1 = n \\ 0, & i \neq 1, x_1 \neq n \\ Nc_n - 1, & i = 1, x_1 = n \\ Nc_n, & i = 1, x_1 \neq n \end{cases} \tag{5.177}$$

where $c_n \coloneqq C(x)_n$ again denotes the global share of opinion n in the system. Thus, the propensity functions are

$$\frac{1}{N} \alpha_{m \to n}(x) \coloneqq \frac{1}{N} \sum_{i : x_i = m} r_{m,n} \frac{d_{i,n}(x)}{d_i} + \tilde{r}_{m,n} \tag{5.178}$$

$$= \begin{cases} r_{m,n} c_m + \tilde{r}_{m,n} c_m, & x_1 = n \\ r_{m,n} \frac{c_n}{N-1} + \tilde{r}_{m,n} c_m, & x_1 \neq n \end{cases} \tag{5.179}$$

which leads to the following reduced propensity functions as $N \to \infty$

$$\tilde{\alpha}_{m \to n}(c) = \begin{cases} r_{m,n} c_m + \tilde{r}_{m,n} c_m, & x_1 = n \\ \tilde{r}_{m,n} c_m, & x_1 \neq n. \end{cases} \tag{5.180}$$

Note that the above equation is not closed due to the dependence on the state x_1 of the central node. Hence, even as the population N goes to infinity, the central node influences the macroscopic dynamics significantly whereas the impact of each single ordinary node vanishes.

In the case of $M = 2$ opinions, which will be denoted as 0 and 1, the resulting infinitesimal change of the share c_0 of opinion 0 is thus given by

$$\frac{\mathrm{d}}{\mathrm{d}t} c_0(t) = \tilde{\alpha}_{1 \to 0}\big(c(t)\big) - \tilde{\alpha}_{0 \to 1}\big(c(t)\big) \tag{5.181}$$

$$= \begin{cases} c_1(t)(r_{1,0} + \tilde{r}_{1,0}) - c_0(t)\tilde{r}_{0,1}, & x_1 = 0 \\ c_1(t)\tilde{r}_{1,0} - c_0(t)(r_{0,1} + \tilde{r}_{0,1}), & x_1 = 1. \end{cases} \tag{5.182}$$

It is clear that, between the stochastic jump events of the central node x_1, the above equation constitutes a deterministic mean-field limit. As a consequence, the collective variables used to describe the macroscopic dynamics are required to not only contain the shares c of each opinion but also the state x_1 of the central node. The associated reduced system hence contains the stochastic jumps of the central node in addition to a deterministic mean-field limit describing the large population of ordinary nodes.

In the following a modification of the CNVM is presented that allows a more refined modeling of leaders and followers.

The Leader-Follower CNVM. Written in the framework presented in the beginning of this chapter, consider $K = 2$ classes that will be denoted as F for follower and L for leader, e.g., a node in extended state (m, F) is a follower with opinion m. Recall also that classes are fixed, i.e., leaders stay leaders and followers stay followers, but their opinions m change over time. The rate at which the opinion of node i changes depends on its class s_i. Consider the following rates that can be seen as an extension of the continuous-time noisy voter model (CNVM):

$$\big(Q_i(x)\big)_{m,n} := \begin{cases} \tilde{r}^F_{m,n} + r^{F,F}_{m,n}\dfrac{d_{i,n,F}(x)}{d_i} + r^{L,F}_{m,n}\dfrac{d_{i,n,L}(x)}{d_i}, & s_i = F \\ \tilde{r}^L_{m,n} + r^{L,L}_{m,n}\dfrac{d_{i,n,L}(x)}{d_i} + r^{F,L}_{m,n}\dfrac{d_{i,n,F}(x)}{d_i}, & s_i = L \end{cases} \tag{5.183}$$

where $d_{i,n,k}(x)$ is the number of neighbors of node i that have opinion n and class k, and d_i is the degree of node i. Each class k is exposed to a specific amount of noise $\tilde{r}^k$ and has specific imitation rates $r^{k,k}$ for copying the opinion of members of their own class. Imitation across classes is dictated by the rates $r^{L,F}$ and $r^{F,L}$, e.g., a follower imitates a leader in its neighborhood at the rate specified in $r^{L,F}$.

It is assumed that in the large population limit $N \to \infty$ the number of leaders N_L is constant and finite, $N_L = \mathcal{O}(1)$, while the number of followers N_F grows linearly, $N_F = \mathcal{O}(N)$. This is a distinctive difference to the heterogeneous population considered in section 5.2.2, where it was assumed that each class makes up a non-vanishing share of the total population. As a consequence of the finite number of leaders, the collective variable $C(x)$ has to be modified. In the original setting $C(x)$ measures the share of each extended state (m, k) in the system, such that the impact of the finite number of leaders would vanish for large N. Hence, the opinions of leaders will be counted in absolute numbers, while the opinions of followers remain as shares, i.e.,

$$C_{(m,k)}(x) := \begin{cases} \dfrac{\#\{i \in [N] : (x_i, s_i) = (m, k)\}}{N}, & k = F \\ \#\{i \in [N] : (x_i, s_i) = (m, k), & k = L. \end{cases} \tag{5.184}$$

Analogously to the original setting, the projected process $\boldsymbol{c}(t) := C(\boldsymbol{x}(t))$ can then be written as

$$\boldsymbol{c}(t) = \boldsymbol{c}(0) + \sum_{m\to n} \Big[\boldsymbol{P}_{(m,L)\to n}\Big(\int_0^t \alpha_{(m,L)\to n}(\boldsymbol{x}(s))\,\mathrm{d}s\Big)\big(e_{(n,L)} - e_{(m,L)}\big) \tag{5.185}$$

$$+\boldsymbol{P}_{(m,F)\to n}\Big(\int_0^t \alpha_{(m,F)\to n}(\boldsymbol{x}(s))\,\mathrm{d}s\Big)\frac{1}{N}\big(e_{(n,F)} - e_{(m,F)}\big)\Big], \tag{5.186}$$

where the $\boldsymbol{P}_{(m,k)\to n}$ are independent Poisson processes and the propensity functions $\alpha_{(m,k)\to n}$ calculate the cumulative rate of the transition $(m,k) \to n$ in the system. Now, assume again the existence of reduced propensity functions $\tilde{\alpha}$ that only depend on the projected state, i.e.,

$$\frac{1}{N}\alpha_{(m,F)\to n}(x) \approx \tilde{\alpha}_{(m,F)\to n}(C(x)) \tag{5.187}$$

$$\alpha_{(m,L)\to n}(x) \approx \tilde{\alpha}_{(m,L)\to n}(C(x)). \tag{5.188}$$

Note that due to the finite number of leaders the reduced propensities $\tilde{\alpha}_{(m,L)\to n}$ do not have to be scaled with N. By the same concentration arguments as before, this implies that the shares of the large number of followers show an almost deterministic behavior that is characterized by a mean-field equation. The states of the leaders however remain a stochastic quantity due to their low number. As a result, the limiting behavior of the system is characterized by a piecewise-deterministic Markov process (PDMP) [152]

$$\begin{aligned}\boldsymbol{c}(t) = \boldsymbol{c}(0) &+ \sum_{m\to n} \boldsymbol{P}_{(m,L)\to n}\Big(\int_0^t \tilde{\alpha}_{(m,L)\to n}(\boldsymbol{c}(s))\,\mathrm{d}s\Big)\big(e_{(n,L)} - e_{(m,L)}\big)\\ &+ \sum_{m\to n}\int_0^t \tilde{\alpha}_{(m,F)\to n}(\boldsymbol{c}(s))\big(e_{(n,F)} - e_{(m,F)}\big)\mathrm{d}s.\end{aligned} \tag{5.189}$$

consisting of a stochastic part that describes the leaders and a deterministic part describing the mean-field behavior of the followers. The resulting trajectories are deterministic in the time intervals between the random transition events of the leaders. Because leaders and followers are modeled differently, the above approximation is a *hybrid model*.

For the PDMP (5.189) to provide a meaningful approximation, the leader nodes have to have an impact in the large population limit even though their number is vanishingly small by comparison. This can for example be achieved by linearly scaling their influence $r^{L,F}$ on followers with N. However, if it is desired that all rate matrices remain constant with respect to N, as was always the case in this chapter, it is easy to see that the node degrees of followers have to stay bounded. If they grew unrestrictedly, the influence of leaders on followers vanishes due to the finite number of leaders, see equation (5.183), making them irrelevant in the model.

Example. A uniformly random d-regular graph with N_F nodes is sampled as in section 5.2.4, and all nodes get assigned to the follower class. Then, N_L leader nodes are added and connected to all other nodes. This yields the reduced propensity functions

$$\tilde{\alpha}_{(m,F)\to n}(c) = c_{(m,F)}\Big(r^F_{m,n} + r^{F,F}_{m,n}c_{(n,F)} + r^{L,F}_{m,n}\frac{c_{(n,L)}}{d}\Big) \tag{5.190}$$

$$\tilde{\alpha}_{(m,L)\to n}(c) = c_{(m,L)}\Big(r^L_{m,n} + r^{L,L}_{m,n}\frac{c_{(n,L)}}{N} + r^{F,L}_{m,n}c_{(n,F)}\Big). \tag{5.191}$$

Hence, in the time interval between two jumps of leaders the shares c approximately evolve according to the mean-field equation (MFE)

$$\frac{\mathrm{d}}{\mathrm{d}t}c(t) = \sum_{m\to n}\tilde{\alpha}_{(m,F)\to n}(c(t))\big(e_{(n,F)} - e_{(m,F)}\big) \tag{5.192}$$

$$= \sum_{m\to n} c_{(m,F)}\Big(r^F_{m,n} + r^{FF}_{m,n}c_{(n,F)} + r^{LF}_{m,n}\frac{c_{(n,L)}}{d}\Big)\big(e_{(n,F)} - e_{(m,F)}\big). \tag{5.193}$$

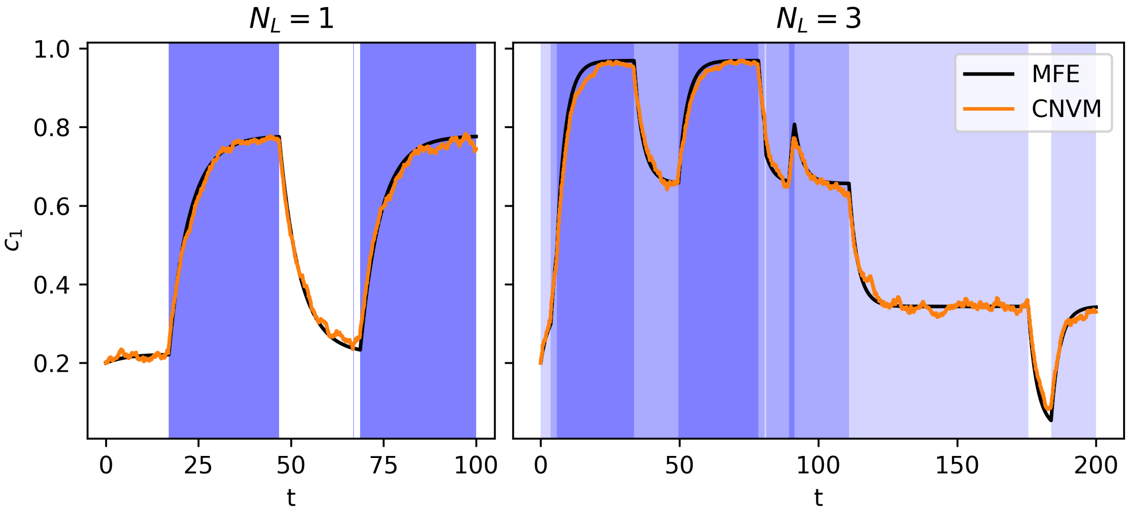


Figure 5.8.: Single trajectory of the CNVM with N_L leaders on a 10-regular graph. The background color describes the states of the leaders and c_1 the share of followers with opinion 1. The MFE (5.193) is used to approximate c between the jumps of leaders.

As was shown in section 5.2.4, the MFE can only become exact in the large population limit if the degree d grows unbounded, which is explicitly not allowed here. Thus, in this case the above PDMP and MFE are only approximations, not exact limits. However, it was also described earlier that the larger d is, the better the mean-field approximation becomes. Hence, the error of the approximation can be controlled by increasing d.

Consider the parameters $d = 10$, $N_F = 5000$, $M = 2$, and the rates

$$\tilde{r}^F_{m,n} = \tilde{r}^L_{m,n} = 0.01, \quad r^{F,F}_{m,n} = r^{L,F}_{m,n} = 1, \quad r^{L,L}_{m,n} = r^{F,L}_{m,n} = 0. \tag{5.194}$$

In this setup, the leaders only change their opinions due to noise $\tilde{r}^L$. Figure 5.8 shows that the MFE (5.193) is indeed a good approximation between the jumps of leaders. After a single trajectory of the CNVM has been simulated, the time points of transitions of leaders and their new states are used to solve the MFE in between. Note that the initial condition for the MFE is not reset after each leader jump, but the last value for c of the last interval is used as the initial value in the next interval, leading to a continuous trajectory. Still, even after several jumps of leaders the MFE matches the simulated trajectory closely.

However, this experiment does not show if the jumps of leaders are distributed correctly in the PDMP (5.189), which depends on the approximation error of the reduced propensities $\tilde{\alpha}_{(m,L)}$. A more thorough investigation of the PDMP, including a detailed analysis of the errors, could be the subject of future work.

6. Learning collective variables

The shares of each state in a discrete-state dynamical system on a network are appropriate collective variables (CVs) for large, dense and homogeneous networks, which was discussed in detail in the previous chapter. In the case of more complicated heterogeneous or sparse networks, other choices of CVs are often necessary to describe the macroscopic behavior. As finding good CVs purely analytically is an intricate task, or even infeasible for very complex systems, and typically involves expert knowledge about the emergent mechanisms combined with some guesswork, this chapter focuses on a data-driven method instead. This data-driven method for automatically learning good collective variables for discrete-state dynamical systems on networks is based on the transition manifold approach, which is introduced in section 6.1. The extension of this approach to dynamical systems on networks is presented in section 6.2, including several examples and a technique for the validation of the results. The CVs learned by this method agree well with the literature in cases where good CVs are already known, but novel results for cases that have not been solved analytically yet are also presented.

While the method is able to provide good collective variables, it does not generate a reduced system. However, to find the reduced system, for example in the form of a differential equation, existing data-driven techniques for learning dynamics can be appended to form a complete model reduction pipeline. This is demonstrated in section 6.3, where a reduced system for the voter model on Albert–Barabási networks is learned based on CVs that are derived by the transition manifold approach.

Section 6.2, i.e., the presentation of the method and the application to examples, is largely adopted from the author's publication [2].

6.1. The transition manifold approach

The *transition manifold approach* [56, 57, 58] allows the data-driven calculation of collective variables (CVs) by examining the geometry of the set of transition density functions. Given a series of short burst simulations of the underlying system started in different initial states, the distances between transition density functions can be estimated, which enables the derivation of optimal CVs. In comparison to black-box learning algorithms utilizing neural networks, the transition manifold approach has the advantage that explicit dynamical conditions for the existence of good CVs can be stated and validated during the data-driven computation, making the resulting CVs more comprehensible and interpretable.

Let a Markovian stochastic process $\boldsymbol{x}(t)$ on the compact connected state space $\mathbb{X} \subset \mathbb{R}^N$ be given. Assume that for each $x \in \mathbb{X}$ and $t > 0$ the *transition density function* $p_x^t : \mathbb{X} \to \mathbb{R}_{\geq 0}$ can be defined as the density (w.r.t. Lebesgue) of the distribution of $\boldsymbol{x}(t)$ given $\boldsymbol{x}(0) = x$. Furthermore, assume that the stochastic process is reversible and ergodic such that it exhibits a unique stationary distribution with density ρ. The *transition manifold* *approach* works by examining the set

$$\mathbb{M}_\tau := \{p_x^\tau \mid x \in \mathbb{X}\} \subset L^1(\mathbb{X}) \tag{6.1}$$

for an appropriate *lag-time* τ. If τ is very small compared to the relevant timescales of the system, the transition density functions are close to Dirac distributions, $p_x^\tau \approx \delta_x$ for all x. Hence, the set $\mathbb{M}_\tau$ is clearly one-to-one to the state space $\mathbb{X}$, i.e., it is an up to N-dimensional manifold. In the limit of large lag-times $\tau \to \infty$ on the other hand, all p_x^τ converge to the stationary distribution $\rho \in L^1(\mathbb{X})$ of the process. As a consequence, for very large lag-times compared to the relevant timescales it is essentially $\mathbb{M}_\tau \approx \{\rho\}$, i.e., a 0-dimensional space. Thus, as one lets the lag-time increase from small to large, the set $\mathbb{M}_\tau$ changes from a potentially N-dimensional object to essentially a single point.

It has been observed that, if the stochastic process can be reduced to a smaller d-dimensional system as described in section 2.3, the set $\mathbb{M}_\tau$ (for appropriate τ) will be close to a d-dimensional submanifold $\mathbb{M} \subset L^1(\mathbb{X})$ called the *transition manifold*, i.e., there exists a small $\varepsilon > 0$ with

$$\inf_{f\in\mathbb{M}} \|f - p_x^\tau\| \leq \varepsilon \tag{6.2}$$

for all $x \in \mathbb{X}$. This phenomenon is robust in the sense that $\mathbb{M}_\tau$, on its journey from N-dimensional to 0-dimensional, will be close to a d-dimensional manifold for a wide range of lag-times τ.

The reason for this observation is that, if good collective variables $\varphi : \mathbb{X} \to \mathbb{R}^d$ exist such that (compare eq. (2.12))

$$p_x^t \approx \tilde{p}_{\varphi(x)}^t \tag{6.3}$$

for all x, then $\mathbb{M}_\tau$ is "almost parametrizable" in d coordinates. On the other hand, if there exists a d-dimensional transition manifold $\mathbb{M}$ fulfilling (6.2), it can be shown [57, theorem 2.5] that good d-dimensional CVs are given by a parametrization of that manifold. To specify this, consider the projection Q onto $\mathbb{M}$,

$$Q : \mathbb{X} \to \mathbb{M}, \quad Q(x) := \operatorname{argmin}_{f\in\mathbb{M}} \|f - p_x^\tau\|. \tag{6.4}$$

Furthermore, assume that an injective parametrization $\gamma : \mathbb{M} \to \mathbb{R}^d$ of the transition manifold is known. Then the CV $\varphi : \mathbb{X} \to \mathbb{R}^d$, $\varphi := \gamma \circ Q$, is good in the sense that it is able to parametrize the dominant spectrum of the transfer operator of the process $\boldsymbol{x}(t)$. More precisely, let $\theta_i : \mathbb{X} \to \mathbb{R}$ denote the eigenfunction of the i-th largest eigenvalue λ_i of the transfer operator. Then there exists a function $\tilde{\theta}_i : \mathbb{R}^d \to \mathbb{R}$ such that

$$\|\theta_i - \tilde{\theta}_i \circ \varphi\| \leq \frac{\varepsilon}{|\lambda_i|}. \tag{6.5}$$

Thus, especially the eigenfunctions of the dominant spectrum are well represented by the CV. As explained in section 2.3, this allows the definition of a reduced effective dynamical system that closely replicates the original dynamics, i.e., the CVs are *good*.

Approximating the transition manifold from data. As the theoretical considerations outlined above suggest, collective variables could be obtained in practice by learning a parametrization of the transition manifold $\mathbb{M} \subset L^1$ from data. Because the transition densities p_x^τ are infinite dimensional objects, this endeavor seems futile at first. However, it is not necessary to explicitly calculate the p_x^τ. Instead, it is sufficient to estimate the distances between them, which can be achieved efficiently using a kernel-based approach presented in [57]. The distance metric employed is the *maximum mean discrepancy* (MMD) [153], which measures the difference between the means of two distributions after mapping them into a reproducing kernel Hilbert space $\mathbb{H}$ that is also called the *feature space*. The feature space is induced by a positive definite kernel function $\kappa : \mathbb{X} \times \mathbb{X} \to \mathbb{R}$. This kernel κ also induces a *feature map* $\phi : \mathbb{X} \to \mathbb{H}$ such that

$$\kappa(x, y) = \langle \phi(x), \phi(y) \rangle_{\mathbb{H}}. \tag{6.6}$$

The MMD between two transition densities p_x^τ and p_y^τ is then defined as

$$\mathrm{MMD}^2\big(p_x^\tau, p_y^\tau\big) := \; \big\| \mathbb{E}\big[\phi\big(\boldsymbol{x}(\tau, x)\big)\big] - \mathbb{E}\big[\phi\big(\boldsymbol{x}(\tau, y)\big)\big] \big\|_{\mathbb{H}}^2, \tag{6.7}$$

where $\boldsymbol{x}(\tau, x) \sim p_x^\tau$ denotes the stochastic process at time τ when starting in x at time 0. Here, $\mathbb{E}\big[\phi\big(\boldsymbol{x}(\tau, x)\big)\big] = \int \phi(z) p_x^\tau(dz)$ is a Hilbert-space valued integral. Its computation is, however, not required for the evaluation of the MMD: Given (6.6), the definition of the MMD can be rewritten using the kernel, which is often referred to as the “kernel trick”

$$\begin{aligned}\mathrm{MMD}^2\big(p_x^\tau, p_y^\tau\big) = \; & \mathbb{E}\Big[\kappa\big(\boldsymbol{x}(\tau, x), \tilde{\boldsymbol{x}}(\tau, x)\big)\Big] + \mathbb{E}\Big[\kappa\big(\boldsymbol{x}(\tau, y), \tilde{\boldsymbol{x}}(\tau, y)\big)\Big] \\ & - 2 \, \mathbb{E}\Big[\kappa\big(\boldsymbol{x}(\tau, x), \tilde{\boldsymbol{x}}(\tau, y)\big)\Big],\end{aligned} \tag{6.8}$$

where $\boldsymbol{x}(\tau, x), \tilde{\boldsymbol{x}}(\tau, x) \sim p_x^\tau$ and $\boldsymbol{x}(\tau, y), \tilde{\boldsymbol{x}}(\tau, y) \sim p_y^\tau$ are stochastically independent. This allows the approximation of the MMD by averaging over kernel function evaluations at samples of p_x^τ and p_y^τ.

Let a set of initial states $x^1, \ldots, x^K \in \mathbb{X}$, which are called the *anchor points* in the following, be given. Define the distance matrix $\Delta \in \mathbb{R}^{K \times K}$ as the matrix containing the squared MMD associated to all pairs of anchor points, i.e.,

$$\Delta_{k_1, k_2} := \mathrm{MMD}^2\big(p_{x^{k_1}}^\tau, p_{x^{k_2}}^\tau\big). \tag{6.9}$$

To estimate Δ, perform $S \in \mathbb{N}$ simulations of duration τ starting in each anchor point x^k and denote the end points of these S simulations by $y^{(k,1)}, \ldots, y^{(k,S)} \in \mathbb{X}$. Then construct the kernel matrix $M \in \mathbb{R}^{K \times K}$ with

$$M_{k_1, k_2} := \frac{1}{S^2} \sum_{s_1, s_2 = 1}^{S} \kappa\big(y^{(k_1, s_1)}, y^{(k_2, s_2)}\big), \tag{6.10}$$

which yields the estimation

$$\Delta_{k_1, k_2} \approx M_{k_1, k_1} + M_{k_2, k_2} - 2 \, M_{k_1, k_2}. \tag{6.11}$$

Finally, apply a distance-based manifold learning algorithm to the distance matrix Δ. In the following, the diffusion maps method [154] is employed for this purpose. This yields an approximation to the d-dimensional CV φ evaluated at the anchor points x^k, i.e., the output is the evaluations $\varphi(x^1), \ldots, \varphi(x^K) \in \mathbb{R}^d$.

Further considerations. Regarding the choice of kernel κ in the above algorithm, it is recommended to use a kernel that is *characteristic* [153], which implies that the MMD is indeed a metric, i.e., it is zero if and only if the distributions are exactly equal. A popular example is the *Gaussian kernel*

$$\kappa(x, y) := \exp\Big(- \frac{\|x-y\|_2^2}{\sigma^2} \Big), \tag{6.12}$$

where the parameter σ is called the *bandwidth*.

Furthermore, an appropriate choice of anchor points $x^1, \ldots, x^K \in \mathbb{X}$ is crucial for the success of the method. It is required that they are sufficiently diverse in the sense that their respective transition densities cover $\mathbb{M}_\tau$ well. Otherwise, the learned parametrization would be biased by the insufficient coverage and thus could fail to produce a good CV for the entire process. A precise quantification of sufficient coverage cannot be stated in generality as it depends on the system at hand and on the desired quality of the CVs.

The manifold learning technique called *diffusion maps*, which was mentioned above, works by applying a kernel to the distance matrix Δ to estimate local similarity between data points. A popular choice is again the Gaussian kernel, i.e., the local similarity between points k_1 and k_2 is evaluated as $\exp(-\Delta_{k_1,k_2}/\tilde{\sigma}^2)$. Here an optimal bandwidth $\tilde{\sigma}$ can actually be inferred from the distances Δ, see [51, appendix B] for details. In the next step, a Fokker–Planck diffusion is employed to construct a random walk on these points such that the jump probabilities between points are proportional to their local similarity calculated above. (This means the diffusion maps normalization parameter is $\alpha = 1/2$ in [154].) A low-dimensional representation of the data is then obtained from the dominant eigenvectors of the resulting diffusion matrix. By examining the distance matrix and the dominant spectrum of the diffusion matrix it is possible to deduce a reasonable estimate of the dimension d of the transition manifold [155, 51].

6.2. Interpretable collective variables for dynamical systems on networks

In this section the transition manifold approach described above is extended in order to learn CVs for discrete-state dynamical systems on networks and to systematically find the relationship between the learned CVs and topological features of the network. To make the method applicable to such processes on networks, a technique for evenly sampling anchor points is developed and a linear regression step is added to make the learned CVs more interpretable. To demonstrate the method, it is then applied to several examples: a clustered network generated by the stochastic block model in section 6.2.1, a ring graph in section 6.2.2, a random regular network in section 6.2.3, and an Albert–Barabási network in section 6.2.4. Finally, a technique to validate the quality of the calculated CVs is presented in section 6.2.5.

Consider again a fixed network of N nodes, on which each node $i \in [N]$ has a discrete state $x_i \in [M]$. The state space of the process is thus $\mathbb{X} := [M]^N$, and its elements are system states $x = (x_1, \ldots, x_N)$. Moreover, let some Markovian dynamics be given that dictates the stochastic evolution of the system. As a consequence of the discrete state space, the transition density functions p_x^τ are technically not density functions but probability vectors, i.e., $p_x^\tau(y)$ is the probability that the system is in state y at time τ

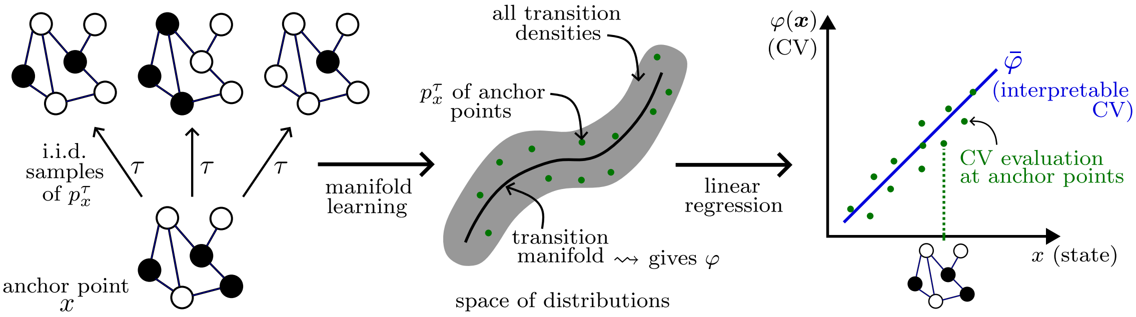


Figure 6.1.: Illustration of the method. Left: The random process is described by its distribution, sampled through S samples per initial network state (anchor point). Middle: These distributions are used to learn a low-dimensional parametrization of the transition manifold. Right: A regression step allows for interpretability of the learned CV. Reprinted from [2].

after having started in state x at time 0. Thus, each p_x^τ is an M^N-dimensional probability vector. Nevertheless, even though the original theory is for continuous state spaces, the transition manifold approach outlined in the previous section can easily be translated to the discrete state setting. For instance, while the set $\mathbb{M}_\tau$ is now a finite subset of $\mathbb{R}^{M^N}$, the transition manifold as a submanifold of $\mathbb{R}^{M^N}$ can still be defined as in (6.2), and the subsequently discussed considerations regarding the existence of good collective variables still apply[1]. In the following, only binary-state dynamics, i.e., $M = 2$ and $x_i \in \{0, 1\}$, will be considered. The method presented below could certainly be extended to an arbitrary number of states but at the cost of significant additional technicalities, which will be addressed further at the end of the section.

The method consists of three steps, which are explained in detail below and are summarized in figure 6.1. The first step addresses the sampling of anchor points. In the second step, the kernel-based approach discussed in the previous section is applied to learn a parametrization of the transition manifold. Finally, in the third step the learned collective variable is extended to unseen states via linear regression, making it easier to comprehend and interpret.

Step 1: sampling anchor points. The aim is to choose a diverse set of dynamically relevant anchor points $x^1, \dots, x^K \in \mathbb{X}$ in which the CV is going to be computed in the first instance. Diversity of the points is crucial in the sense that their respective transition densities have to cover $\mathbb{M}_\tau$ sufficiently well. For example, simply picking random states $x^k \sim \mathrm{Unif}(\{0, 1\}^N)$ does often not produce a desirable set of anchor points, as mostly states with about 50% 1's are sampled (by the law of large numbers). Thus, no insights would be provided on the behavior of the system in states that have substantially more or less 1's than 50%. Instead, the idea is to construct states x that contain communities of nodes with the same state because these are especially dynamically stable for most types of spreading processes. For instance, in an SI model a community of almost exclusively susceptible nodes is "stable" in the sense that it is not changing quickly over time, whereas a community consisting of half susceptible and half infectious nodes would likely

[1] Although, strictly speaking the proof in [57] that the parametrization of the transition manifold is a good CV does not apply to the CNVM because it is not reversible.

quickly become mostly infectious. Thus, as configurations of nodes that involve many alternating states tend to dissolve quickly under most spreading dynamics, the system is predominantly observed in states with uniform clusters (if there are clusters), and the best understanding of long term dynamics is extracted by putting emphasis on such more relevant states.

In order to sample such a set of anchor points, algorithm 5 is employed. It works as follows. Start with an empty (uninitialized) state x and assign some random *seed nodes* to be of state 0 and 1. Then, iteratively assign the state 0 to neighbors of nodes with state 0, and 1 to neighbors of nodes with state 1, until a random target count $c \in \{0, ..., N\}$ of 1's and $N - c$ of 0's is reached. For each sample x^k a random number of seeds between 1 and $N_{\text{seed}} = 5$ is used. Larger or more intricate networks may require a bigger number of seeds. The purpose of the random target counts is that states with all possible counts of 1's and 0's are sampled.

Algorithm 5 Sampling an anchor point x

```
$x \leftarrow$ empty array of size $N$
$c \leftarrow$ sample Unif($\{0, \dots, N\}$)                      ▷ target count of state 1
$n \leftarrow$ sample Unif($\{1, \dots, N_{\text{seed}}\}$)                   ▷ number of seeds
$n \leftarrow \min\{n, c, N - c\}$                    ▷ reduce num. of seeds (if necessary)
$i_1, \dots, i_{2n} \leftarrow$ random indices from $\{1, \dots, N\}$
$x[i_1], \dots, x[i_n] \leftarrow 0$                                ▷ initialize seed points
$x[i_{n+1}], \dots, x[i_{2n}] \leftarrow 1$                             ▷ initialize seed points
while (count of state 0) $< N - c$ or (count of state 1) $< c$ do
    if (count of state 0) $< N - c$ then
        $i \leftarrow$ index of random uninitialized neighbor of a node with state 0
        (If no such $i$ exists, pick any random uninitialized node)
        $x[i] \leftarrow 0$
    end if
    if (count of state 1) $< c$ then
        $i \leftarrow$ index of random uninitialized neighbor of a node with state 1
        (If no such $i$ exists, pick any random uninitialized node)
        $x[i] \leftarrow 1$
    end if
end while
return $x$
```

Step 2: transition manifold. Next, a "parametrization" φ of the transition manifold $\mathbb{M}$ is learned from simulation data using the kernel-based method discussed in the previous section. To this end, $S \in \mathbb{N}$ simulations of length τ are conducted for each anchor point x^k, yielding S samples for each transition density $p^{\tau}_{x^k}$. Using this data, evaluations of the collective variable φ at the anchor points $x^1, \dots, x^K$ are obtained by applying the diffusion maps manifold learning technique to the calculated distance matrix.

The necessary number K of anchor points and S of simulations per anchor point depend on the size and complexity of the network. For the examples that are presented later ($N < 1000$), $K \approx 1000$ and $S \approx 100$ are adequate choices. This very small number of anchor points compared to the number 2^N of all possible states is sufficient due to the targeted sampling method employed in the first step of the method. Furthermore, as many model simulations of length τ must be sampled, it is generally advantageous to choose a rather small value for τ. However, one should be aware that if the lag-time is too small, the fast processes of the system will not equilibrate in that time frame, resulting in an unnecessarily large dimension d and a CV that is too fine-grained. In the case of systems with unknown timescales, it is therefore advisable to examine trajectories to infer a suitable lag-time and to approximate the transition manifold for different τ, comparing their dimensions d. For the CNVM, which will be used in the examples later, a lag-time of the order such that nodes are expected to experience approximately one state transition typically produces satisfactory results, i.e., $\tau \approx (r_{m,n} + \tilde{r}_{m,n})^{-1}$. Nevertheless, minor modifications may be required for specific examples due to effects of the network topology. Although the results of the method exhibit a substantial robustness in varying the above hyperparameters, an entirely automatic procedure for their optimal selection has yet to be designed.

The output of this second step of the method is evaluations of the d-dimensional CV φ at the anchor points, $\varphi(x^1), \dots, \varphi(x^K) \in \mathbb{R}^d$.

Step 3: linear regression. The third step of the method aims at discovering the meaning of the CV and finding a reasonable map $\bar{\varphi} : \{0,1\}^N \to \mathbb{R}^d$ that extends it to states x outside of the original data set. Motivated by the fact that, for binary-state dynamics, the share of nodes in state 1 in (parts of) the network is known to be a good CV in specific cases [1], maps $\bar{\varphi}$ of the form

$$\bar{\varphi}(x) = \begin{pmatrix} \bar{\varphi}_1(x) \\ \vdots \\ \bar{\varphi}_d(x) \end{pmatrix}, \quad \bar{\varphi}_j(x) = \sum_{i=1}^{N} \Lambda_{j,i}\, x_i, \tag{6.13}$$

where $\Lambda \in \mathbb{R}^{d \times N}$ is a parameter matrix, are considered. For example, choosing $d = 1$ and $\Lambda = (1, \dots, 1)$ yields a map describing the total count of state 1 in the system, i.e., the standard *shares* which were discussed many times earlier in this thesis. In the different context of continuous-state dynamics given by coupled ODEs, a CV similar to (6.13) was also examined in [156, 157].

Optimal parameters Λ are found by employing linear regression to fit $\bar{\varphi}$ to the computed CV values in the anchors, $\varphi(x^1), \dots, \varphi(x^K)$, from step 2. To this end, define the matrix $\Phi := (\varphi(x^1), \dots, \varphi(x^K)) \in \mathbb{R}^{d \times K}$, the matrix $X \in \mathbb{R}^{N \times K}$, $X_{i,k} := x_i^k$, and note that

$$\bar{\Phi} := \big(\bar{\varphi}(x^1), \dots, \bar{\varphi}(x^K)\big) = \Lambda X \in \mathbb{R}^{d \times K}. \tag{6.14}$$

Optimal parameters Λ, that yield a maximal correlation between the fit $\bar{\Phi}$ and the data Φ, are then found by solving the centered linear regression problem for the rows of Λ

$$\Lambda_{j,:} = \underset{\lambda \in \mathbb{R}^N}{\operatorname{argmin}} \| E X^T \lambda - E \Phi_{j,:} \|_2^2, \quad j = 1, \dots, d. \tag{6.15}$$

Here, the operator $E := I - \frac{1}{K}\mathbf{1}(\mathbf{1}^T) \in \mathbb{R}^{K \times K}$, where I denotes the identity matrix and $\mathbf{1} := (1, \dots, 1)^T$, centers a vector around its mean.

To prevent overfitting of the parameters Λ to the data, the *graph total variation* regularizer

$$\mathrm{TV}(\Lambda_{j,:}) \coloneqq \sum_{(i,k)\in\mathcal{E}} |\Lambda_{j,i} - \Lambda_{j,k}|, \tag{6.16}$$

is employed, where $\mathcal{E}$ denotes the edge set of the graph. The reason for this choice of regularizer that penalizes variation of Λ along edges of the graph is that each node in densely interconnected clusters is expected to contribute similarly to the CV. This yields the generalized LASSO [158] regression problem

$$\Lambda_{j,:} = \operatorname*{argmin}_{\lambda\in\mathbb{R}^N} \|EX^T\lambda - E\Phi_{j,:}\|_2^2 + \alpha\cdot\mathrm{TV}(\lambda). \tag{6.17}$$

The strength $\alpha > 0$ of the penalty can be optimized for a given data set via cross-validation [159]. Due to the regularization, the solution $\Lambda_{j,:}$ tends to be constant within network communities, which reinforces the interpretation of $\bar{\varphi}$ to measure the count of 1's in densely connected regions of the network. The optimization problem (6.17) is convex and can be efficiently solved using off-the-shelf solvers. In the examples discussed below, the OSQP solver [160] is used.

For many networks that have been studied, the ansatz functions presented in (6.13) resulted in CVs of good quality. In some cases however, it can be beneficial to pre-weight each node. For example, for networks containing hubs, like Albert–Barabási networks, a strong correlation of the optimal (in the sense of (6.17)) weights Λ with the node degree can sometimes be found. This conflicts with the graph total variation regularizer because the network contains many edges between nodes of substantially different degree. A solution to this issue, which will be demonstrated later for Albert–Barabási networks in section 6.2.4, is to incorporate a pre-weighting of each node with its degree into the ansatz functions:

$$\bar{\varphi}_j(x) = \sum_{i=1}^{N} \Lambda_{j,i}\, d_i\, x_i, \tag{6.18}$$

where d_i denotes the degree of node i. Transferring this pre-weighting to the regression problem (6.17) yields

$$\Lambda_{j,:} = \operatorname*{argmin}_{\lambda\in\mathbb{R}^N} \|EX^T D\lambda - E\Phi_{j,:}\|_2^2 + \alpha\cdot\mathrm{TV}(\lambda), \tag{6.19}$$

where $D \coloneqq \mathrm{diag}(d_1,\dots,d_N) \in \mathbb{R}^{N\times N}$.

If the ansatz functions suggested above still do not yield satisfying results for a particular network or a particular system, they could also be modified using a different pre-weighting or even completely replaced by other ansatz functions that are better suited for that problem. An interesting idea would for example be to complement the suggested ansatz functions with functions that can resolve the counts of network motifs that are used in moment closure methods (see section 2.4.2), e.g., the count of links between nodes in state 0 and in state 1, or the counts of certain triplets.

Extension to more than two states. The steps 1 and 2 of the method can be applied without significant modification to systems where nodes have one of $M \in \mathbb{N}$ discrete states. Step 3 however is specifically tailored to binary-state systems due to the specific choice of ansatz functions (6.13). A straightforward way to generalize these ansatz functions to M states is by checking each node i for its state and then adding a weight depending on that state, i.e., the j-th coordinate of the CV is given by

$$\bar{\varphi}_j(x) = \sum_{i=1}^{N} \sum_{m=1}^{M} \Lambda_{j,i,m} \, \delta_{\{x_i=m\}}(x), \tag{6.20}$$

where $\delta_{\{x_i=m\}}(x) \in \{0,1\}$. For instance, if the j-th coordinate should count the number of nodes with state m, one would set $\Lambda_{j,i,m} = 1$ for all i and $\Lambda_{j,i,n} = 0$ for all $n \neq m$ and all i. It would be interesting to examine these or different ansatz functions for the case $M > 2$ in future work by adapting the linear regression problem and the regularizer. However, in the following examples still only the case $M = 2$ is considered.

Numerical examples. In the subsections below, numerical examples for different types of networks are presented. Each example contains results for the continuous-time noisy voter model (CNVM) and for a threshold model. Recall that in the CNVM a node i transitions from its opinion m to a different opinion n at the rate

$$r_{m,n} \frac{d_{i,n}(x)}{d_i} + \tilde{r}_{m,n}, \tag{6.21}$$

where $d_{i,n}(x)$ denotes the number of neighbors of node i with opinion n, and d_i is the degree of node i, see section 2.2 for details. In every example the same parameters r and $\tilde{r}$ are used:

$$r = \begin{pmatrix} - & 0.99 \\ 1.01 & - \end{pmatrix}, \quad \tilde{r} = \begin{pmatrix} - & 0.005 \\ 0.005 & - \end{pmatrix}. \tag{6.22}$$

In contrast to the voter model, where transition rates are proportional to the share of opinions in the neighborhood, the *threshold model* assumes that a switch to a different opinion only occurs if that opinion is already sufficiently established in the neighborhood. More precisely, in the threshold model a node i transitions from state $m \in \{0,1\}$ to state $n = 1 - m$ at the rate

$$\begin{cases} r_{m,n} + \tilde{r}_{m,n}, & \text{if } \frac{d_{i,n}(x)}{d_i} \geq b_{m,n} \\ \tilde{r}_{m,n}, & \text{else} \end{cases} \tag{6.23}$$

where again $r_{m,n}, \tilde{r}_{m,n} \geq 0$ are rate parameters. The parameter $b_{m,n} \in [0,1]$ is the threshold at which a node changes its opinion from m to n due to the influence of neighbors. Similarly to the voter model, the rates $\tilde{r}_{m,n}$ control the noise in the system. In the examples presented below, the following parameters are used:

$$r_{0,1} = r_{1,0} = 1, \quad \tilde{r}_{0,1} = \tilde{r}_{1,0} = 0.1, \quad b_{0,1} = b_{1,0} = 0.5. \tag{6.24}$$

In many cases (and when not specified in the text) the results obtained for the CNVM and for the threshold model are qualitatively identical, e.g., the same dimension and shape of the transition manifold and the same collective variables. This implies that, even though the CNVM and the threshold model have different dynamics, the relevant macroscopic information about the system state is often the same. The properties of the

Table 6.1.: Parameter values used in the examples.

	stoch. block model	ring	3-regular	Albert–Barabási
num. of nodes N	900	50	500	500
num. of anchors K	2000	2000	1000	1000
samples per anchor S	100	300	100	100
τ (voter model)	2	5	4	4
τ (threshold model)	1	3	8	2
CV dim. d	3	5	1	1

underlying network on the other hand have a large impact on the CVs learned by the method, as the following examples show.

For the calculation of the distance matrix Δ, see (6.11), the Gaussian kernel with bandwidth $\sigma = \sqrt{N/2}$ was used, where N is the number of nodes of the respective network. The other parameter values of the examples are summarized in table 6.1. The experiments were conducted on a 16-core CPU with 32 GB of memory and took less than 20 minutes each, including the simulations of the CNVM (or threshold model), the manifold learning, and the linear regression with cross-validation. The most costly step ($\sim$ 70% of runtime) in these examples is the calculation of the transition manifold parametrization, i.e., the calculation of the distance matrix and application of the diffusion maps algorithm. The *Python* code for all examples is available at `https://github.com/lueckem/spreading-processes-CVs`.

6.2.1. Application to the stochastic block model

In this section a network of $N = 900$ nodes that is constructed using the *stochastic block model* is considered. The network consists of three equal size clusters such that cluster 1 and 2 are densely connected, cluster 1 and 3 are connected only sparsely, and cluster 2 and 3 are not connected at all, see figure 6.2. Based on the theoretical considerations presented in section 5.2.3, the optimal CV is expected to be $d = 3$-dimensional and contain the counts of 1's in each cluster. Recall that this CV is exact in the sense that for $N \to \infty$ the projected dynamics converges to a mean-field limit. Applying the learning method and plotting the resulting CV point cloud $\{\varphi(x^1), \dots, \varphi(x^K)\} \subset \mathbb{R}^3$ yields an approximately cuboid shaped transition manifold. The vertices of this cuboid correspond to extreme states x in which for each cluster either all or no nodes have state 1, see figure 6.2. To discover the meaning of the three coordinates $\varphi_1, \varphi_2, \varphi_3$, the optimal fit according to the regression problem proposed in step 3 of the method is calculated, which yields a collective variable $\bar{\varphi}(x) = \Lambda x$ with optimal parameters Λ shown in figure 6.3. The entries of the first row $\Lambda_{1,:}$ are all approximately equal and thus $\bar{\varphi}_1$ describes the count of 1's in the whole network. The optimal $\Lambda_{2,:}$ is positive and constant within cluster 3 and negative and constant within clusters 1 and 2. Thus, $\bar{\varphi}_2$ calculates how the 1's are distributed between clusters $\{1, 2\}$ and $\{3\}$. (For example, a large positive value of $\bar{\varphi}_2$ indicates that there are many 1's in cluster 3 and few in clusters 1 and 2). Finally, $\Lambda_{3,:}$ is

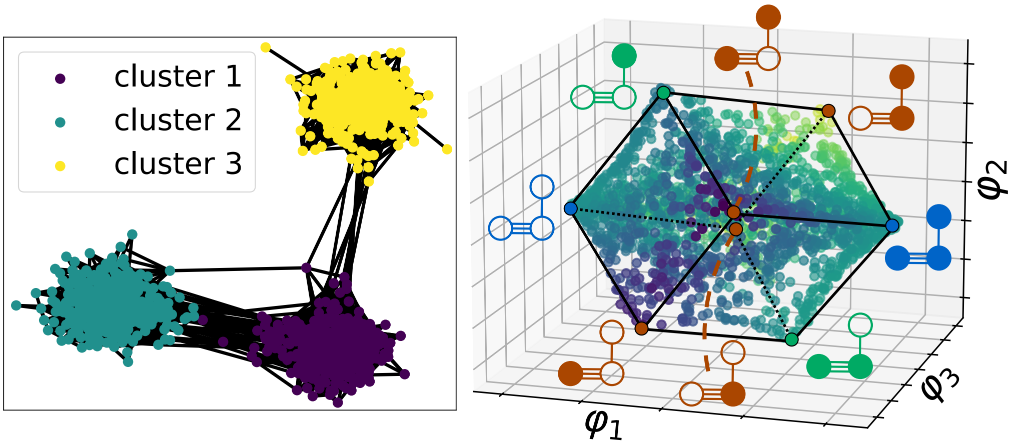


Figure 6.2.: For the stochastic block model network (left), the transition manifold is a 3-dimensional cuboid (right). The vertices of the cuboid correspond to extreme states x where for each cluster either all (filled circle) or no nodes (empty circle) have state 1. Reprinted from [2].

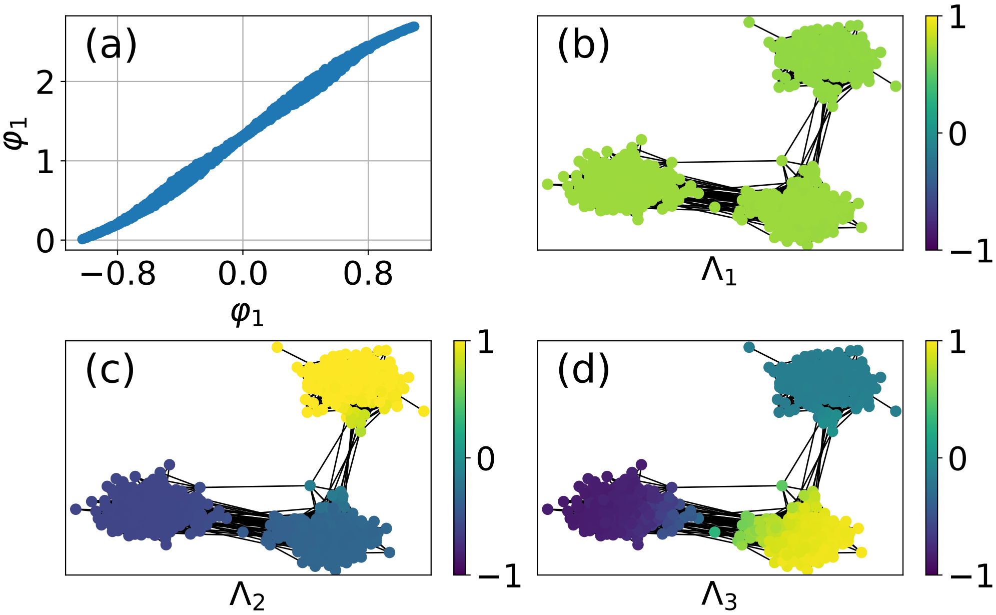


Figure 6.3.: Optimal Λ from (6.13) for the stochastic block model example. (a): Data $\varphi_1(x^k)$ versus optimal fit $\bar{\varphi}_1$. (b)-(d): Optimal Λ entries for the respective coordinates plotted as color values on the network. Reprinted from [2].

positive in cluster 1, negative in cluster 2, and approximately 0 in cluster 3, which implies that $\bar{\varphi}_3$ measures how the 1's are distributed between clusters 1 and 2, regardless of the number of 1's in cluster 3. Hence, the learned CV $\bar{\varphi}$ includes exactly the information that was predicted by the mean-field theory, i.e., the counts of 1's for each cluster, but the coordinates are ordered by dynamical prevalence. For instance, coordinate 3 is the least prevalent because information flows quickly between the two densely connected clusters 1 and 2.

Further numerical examples, which are not shown here, indicate that this result generalizes as the theory predicts, e.g., for a stochastic block model with two clusters a two-dimensional CV is learned were the first coordinate is again the global count of 1's and the second coordinate describes the distribution of 1's between the clusters. For an Erdős–Rényi random graph, i.e., a stochastic block model with only one cluster, the CV is one-dimensional and describes the count of 1's.

An interesting question would be to consider the change of the CV and especially its dimension when increasing the edge density between two clusters. In particular, do structural transitions in the CV coincide with the so-called *detectability threshold* of the stochastic block model [161, 162], where the edge statistics become indistinguishable from an Erdős–Rényi random graph model? This could be addressed in future work.

6.2.2. Application to the ring graph

In this section the method is applied to a ring-shaped network of $N = 50$ nodes. Examining the point cloud $\{\varphi(x^1), \dots, \varphi(x^K)\}$ for different choices of d, a low-dimensional transition manifold can not be identified as increasing d keeps adding important information. To keep the CV dimension reasonably small, a dimension of $d = 5$ is chosen and the larger dimensions are clipped. (This still yields CVs of reasonable quality as will be shown later in section 6.2.5.)

Solving the regression problem in step 3 yields a Λ that is constant in the first coordinate, i.e., the most important information is again the total count of 1's, see figure 6.4. The subsequent $\Lambda_{j,i}$ are pairs of sine and cosine functions of the node index i, starting with one oscillation for coordinates $j = 2, 3$ and then doubling the frequency for coordinates $j = 4, 5$. Hence, the collective variable $\bar{\varphi}$ measures the distribution of 1's on the ring, with increasing precision as d increases. This structure mimics Fourier coefficients, which suggests that (in the limit of infinitely many nodes) the optimal collective variable measures the position-dependent concentration of 1's as a density function on the ring. This result agrees well with other works considering ring-shaped or lattice networks, e.g. [82, 83, 84], which find that the concentration of 1's is governed by a diffusive PDE in the hydrodynamic limit. The CV of the system thus being a function on the ring, any finite-dimensional approximation has a truncation error. However, orthogonal trigonometric polynomials are a natural (and in an L^2-sense optimal) choice, found by our method.

6.2.3. Application to regular networks

A challenge in reduced modeling of spreading processes on random regular graphs is that edges are correlated. If the degree grows indefinitely with the network size, it was shown for the CNVM that the share of state 1 is an asymptotically perfect CV in section 5.2.4. It can be observed numerically that for small degrees this CV still seems to support an ef-

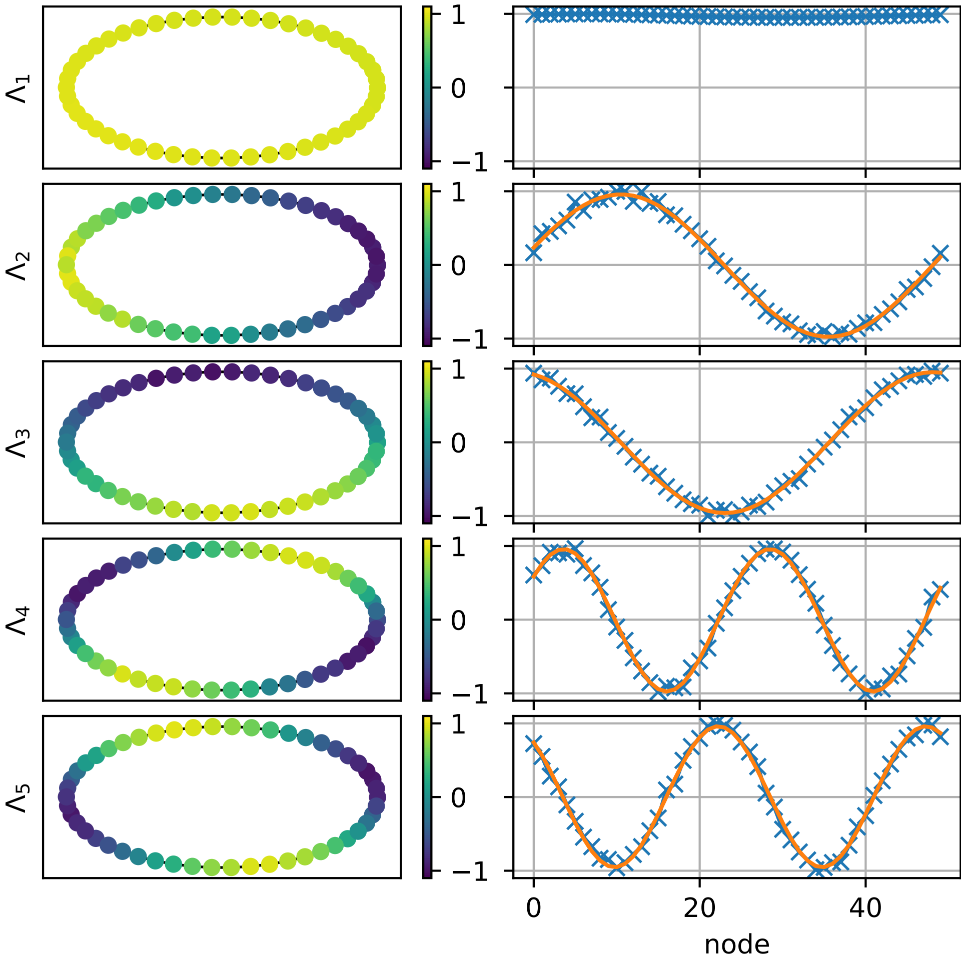
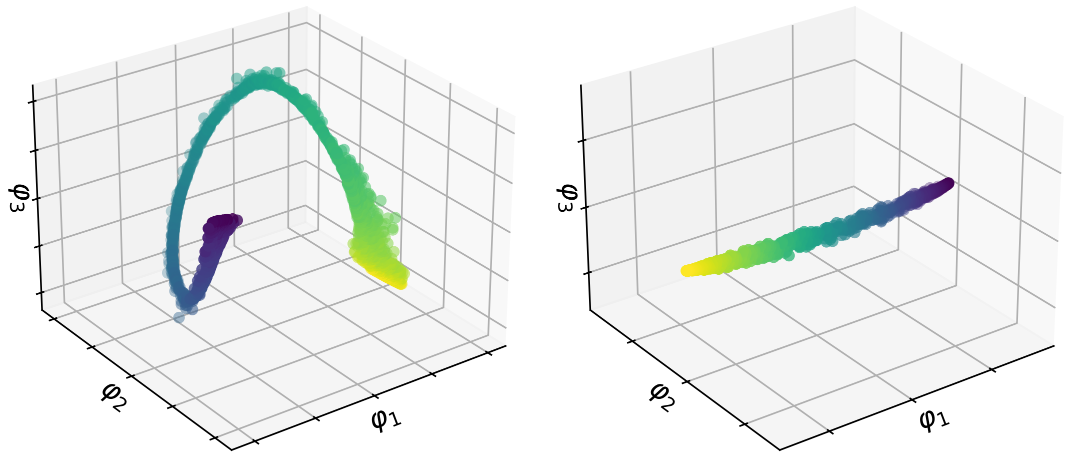


Figure 6.4.: Left: optimal $\Lambda_{i,:}$ plotted as color values on the ring-shaped network. Right: $\Lambda_{i,:}$ (blue crosses) and a sine fit (orange line). The collective variables φ_i represent the real Fourier coefficients of the distribution of 1's on the ring, since $\varphi_i(x) \approx \Lambda_{i,:}\, x$ with the $\Lambda_{i,:}$ being sines and cosines of increasing frequencies. Reprinted from [2].



(a) Voter model.

(b) Threshold model.

Figure 6.5.: For the random 3-regular graph the transition manifold is one-dimensional. The color represents the share of 1's in the associated state.

fective dynamics, but it deviates from the one obtained by mean-field approximation. The CV learning method applied to a random 3-regular network with $N = 500$ nodes validates these observations. The resulting transition manifold is approximately one-dimensional and the learned CV again calculates the global share of 1's, see figure 6.5. Hence, the CV is identical to the optimal CV in the complete graph and Erdős–Rényi random graph setting. However, the dynamics on random 3-regular graphs behaves (quantitatively) differently than on a complete graph. In particular, the known mean-field equation, which constitutes the large population limit of the CNVM on complete networks, is not valid for random regular graphs of small degree. The existence of a dynamical equation similar to the mean-field equation that describes the random regular graph setting is indicated by the one-dimensionality of the transition manifold and could be examined in future work.

6.2.4. Application to Albert–Barabási networks

In this section the method is applied to a network generated by the Albert–Barabási model, see section 3.1.5 for details. In the preferential attachment algorithm each new node is connected to $m = 2$ existing nodes that are randomly picked with probability proportional to their degree. This procedure yields (asymptotically) a scale-free network. Applying the method results in a point cloud $\{\varphi(x^1), \dots, \varphi(x^K)\}$ that indicates a $d = 1$-dimensional transition manifold. The optimal $\Lambda \in \mathbb{R}^N$ according to the linear regression problem in step 3 assigns a large positive weight to nodes of high degree, whereas nodes with small degree have small or even negative weight, as illustrated in figure 6.6. This conflicts with the graph total variation regularizer that favors solutions for which Λ is equal for neighboring nodes. As described earlier, this issue can be tackled by applying a pre-weighting of each node i with its degree d_i:

$$\bar{\varphi}(x) = \sum_{i=1}^{N} \Lambda_i \, d_i \, x_i. \tag{6.25}$$

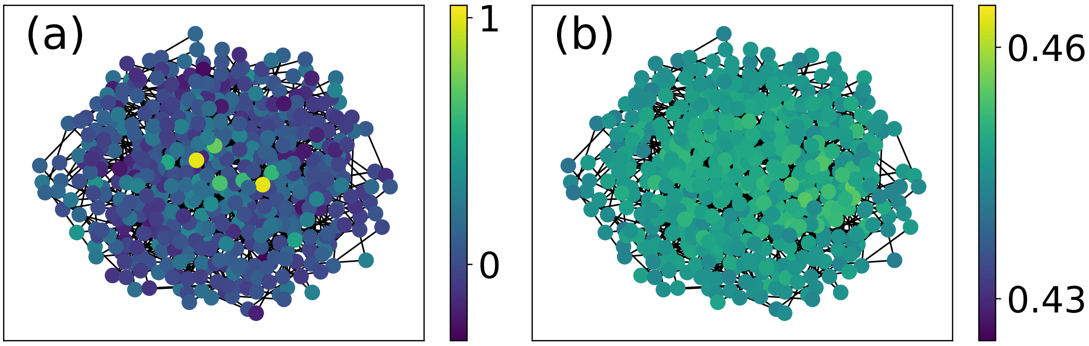


Figure 6.6.: (a) For the Albert–Barabási network, the optimal Λ as in (6.17) assigns a large weight to nodes with high degree. (b) After pre-weighting with node degree, cf. (6.19), the optimal Λ is constant. Hence, the collective variable describes the degree-weighted count of 1's. Reprinted from [2].

As a result, the optimal Λ becomes approximately constant, and hence the CV measures the degree-weighted count of state 1 in the system, see figure 6.6. Multiple experiments for varying parameters confirmed this result, provided the preferential attachment parameter is chosen $m \geq 2$. (For $m = 1$ the resulting networks exhibit a significantly larger diameter. As a consequence, the degree-weighted count does not seem to sufficiently characterize the dynamics.) To the best of the author's knowledge, there are currently no theoretical works showing that the degree-weighted count is a good CV for (binary-state or other) spreading processes on Albert–Barabási networks, although refs. [163, 164] hint at the significance of this observable. A data-driven technique to derive an ODE for the dynamics of the degree-weighted count is presented in section 6.3.

6.2.5. Validation of numerical examples

While the CVs that were learned in the previous examples seem reasonable, it is important to check and validate if they are indeed *good*. Recall that the transition manifold approach is seeking for a low-dimensional parametrization φ of the set of all transition densities p_x^τ, $x \in \{0,1\}^N$. Thus, for two states x and y the distance between p_x^τ and p_y^τ should correlate with the distance between $\varphi(x)$ and $\varphi(y)$. The quality of a collective variable φ can hence be assessed using the following heuristic. Given a small $\varepsilon > 0$ and two states $x^1, x^2 \in \{0,1\}^N$ with $\varphi(x^1) \approx \varphi(x^2)$, define the time t^* as

$$t^* := \inf\{t \geq 0 \mid \forall \tau \geq t : \|p_{x^1}^\tau - p_{x^2}^\tau\| < \varepsilon\}. \tag{6.26}$$

This is well-defined because all p_x^τ are assumed to converge to a unique stationary distribution for $\tau \to \infty$. If t^* is rather large, this implies that the CV is very coarse because initial states with similar CV values may lead to quite different behavior over long timescales. But a good CV has the property that starting the system in x^1 versus in x^2 should make almost no difference after a short time, and hence, good CVs exhibit a small t^*. However, this property alone does not sufficiently characterize the quality of the CV (for example, $\varphi(x) = x$ implies $t^* = 0$, but it does not simplify the state at all). Hence, consider a third state x^3 with a clearly distinct CV value $\varphi(x^3) \not\approx \varphi(x^1)$. Then the CV is of good quality if the associated distributions $p^\tau_{x^1}$ and $p^\tau_{x^3}$ are also substantially different for a long time, i.e.,

$$\hat{t} := \inf\{t \geq 0 \mid \forall \tau \geq t : \|p^\tau_{x^1} - p^\tau_{x^3}\| < \varepsilon\} \tag{6.27}$$

should be large. If $\hat{t}$ is rather small, this implies that the CV is too fine-grained because it assigns different values to initial states that lead to a similar dynamics after a short time. The best possible CV achieves both the smallest t^* for all choices of x^1, x^2 and the largest $\hat{t}$ for all choices of x^1, x^3, as it exactly filters out the state information with a short timescale impact, but keeps the information that leads to fundamentally different dynamics on a long timescale.

The following numerical method can be employed for the validation of CVs. After the approximation $\bar{\varphi}$ of the collective variable has been calculated, pick three states $x^1, x^2, x^3 \in \{0,1\}^N$ as discussed above, i.e, such that $\bar{\varphi}(x^1) \approx \bar{\varphi}(x^2)$ and $\bar{\varphi}(x^3)$ is substantially different.[2] Then compare the distributions $p^\tau_{x^1}$, $p^\tau_{x^2}$, and $p^\tau_{x^3}$ via their maximum mean discrepancy (MMD)

$$\begin{aligned}\mathrm{MMD}^2(x^i, x^j; t) := \; & \mathbb{E}\Big[\kappa\Big(\boldsymbol{x}(t,x^i), \tilde{\boldsymbol{x}}(t,x^i)\Big)\Big] + \mathbb{E}\Big[\kappa\Big(\boldsymbol{x}(t,x^j), \tilde{\boldsymbol{x}}(t,x^j)\Big)\Big] \\ & - 2\,\mathbb{E}\Big[\kappa\Big(\boldsymbol{x}(t,x^i), \tilde{\boldsymbol{x}}(t,x^j)\Big)\Big],\end{aligned} \tag{6.28}$$

where $\boldsymbol{x}(t,x), \tilde{\boldsymbol{x}}(t,x)$ are independent random variables with distribution p^t_x, and $\kappa(\cdot,\cdot)$ is a Gaussian kernel. If the learned CV $\bar{\varphi}$ is good, $\mathrm{MMD}(x^1, x^2; t)$ should decrease to 0 quickly (as t^* is small), while both $\mathrm{MMD}(x^1, x^3; t)$ and $\mathrm{MMD}(x^2, x^3; t)$ stay large for a long time (as $\hat{t}$ is large). The three MMDs are estimated by replacing the expectation in (6.28) with averages from model simulations.

To really prove the quality of the learned CV $\bar{\varphi}$, one would have to conduct this experiment for all possible choices of x^1, x^2, x^3, which is of course not feasible. However, doing it for a few (random) choices is a good indicator of whether the CV is correct.

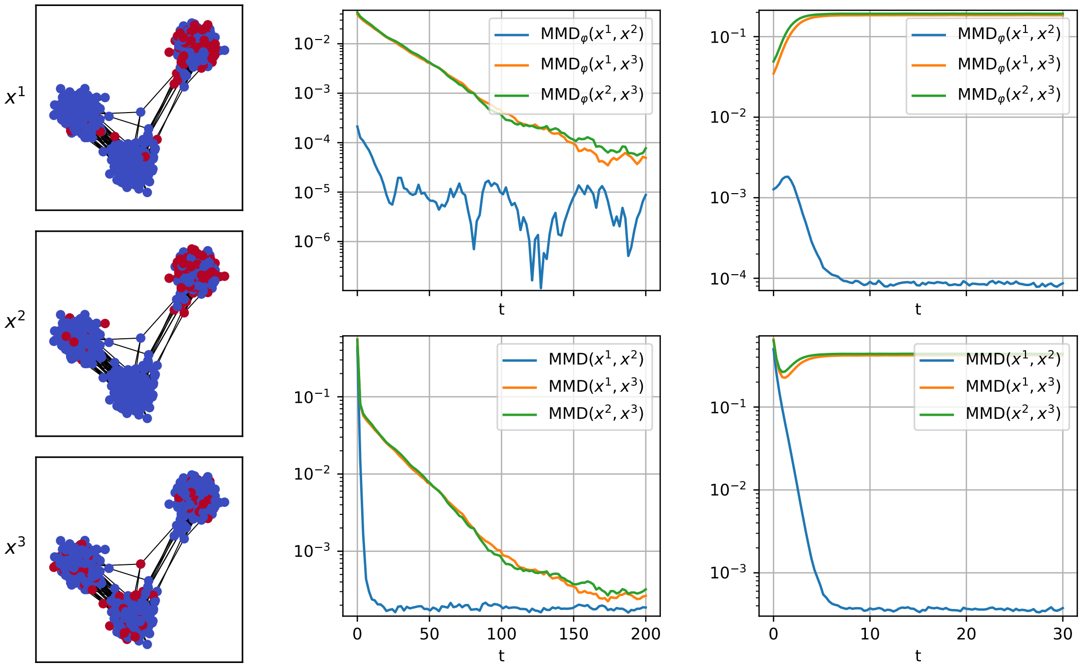
A weaker (but easier to compute) indicator of the quality of a CV is given by the differences between the *projected* distributions, i.e.,

$$\begin{aligned}\mathrm{MMD}^2_\varphi(x^i, x^j; t) := \; & \mathbb{E}\Big[\kappa\Big(\bar{\varphi}(\boldsymbol{x}(t,x^i)), \bar{\varphi}(\tilde{\boldsymbol{x}}(t,x^i))\Big)\Big] + \mathbb{E}\Big[\kappa\Big(\bar{\varphi}(\boldsymbol{x}(t,x^j)), \bar{\varphi}(\tilde{\boldsymbol{x}}(t,x^j))\Big)\Big] \\ & - 2\,\mathbb{E}\Big[\kappa\Big(\bar{\varphi}(\boldsymbol{x}(t,x^i)), \bar{\varphi}(\tilde{\boldsymbol{x}}(t,x^j))\Big)\Big].\end{aligned} \tag{6.29}$$

In contrast to (6.28), $\mathrm{MMD}_\varphi(x^1, x^2; t)$ should be close to 0 for all times t, as states with the same CV value should lead to identical effective (projected) dynamics. The differences $\mathrm{MMD}_\varphi(x^1, x^3; t)$ and $\mathrm{MMD}_\varphi(x^2, x^3; t)$ should again be large for small and intermediate t. For very large t, even they should approach zero due to the convergence to the stationary distribution.

Results. For the stochastic block model example, states x^1 and x^2 were constructed such that the share of 1's is 10% in cluster 1, 0% in cluster 2, and 50% in cluster 3, see figure 6.7. In state x^3 the total number of 1's is the same but they are distributed uniformly, irrespective of the clusters. For both the voter model dynamics and the threshold model

[2] Given x^1, the state x^2 can for instance be sampled using a Markov chain Monte Carlo method. Starting with a uniformly random x^2, randomly flip states of nodes until $\bar{\varphi}(x^1) \approx \bar{\varphi}(x^2)$.



(a) States x^1, x^2, x^3.

(b) Voter model.

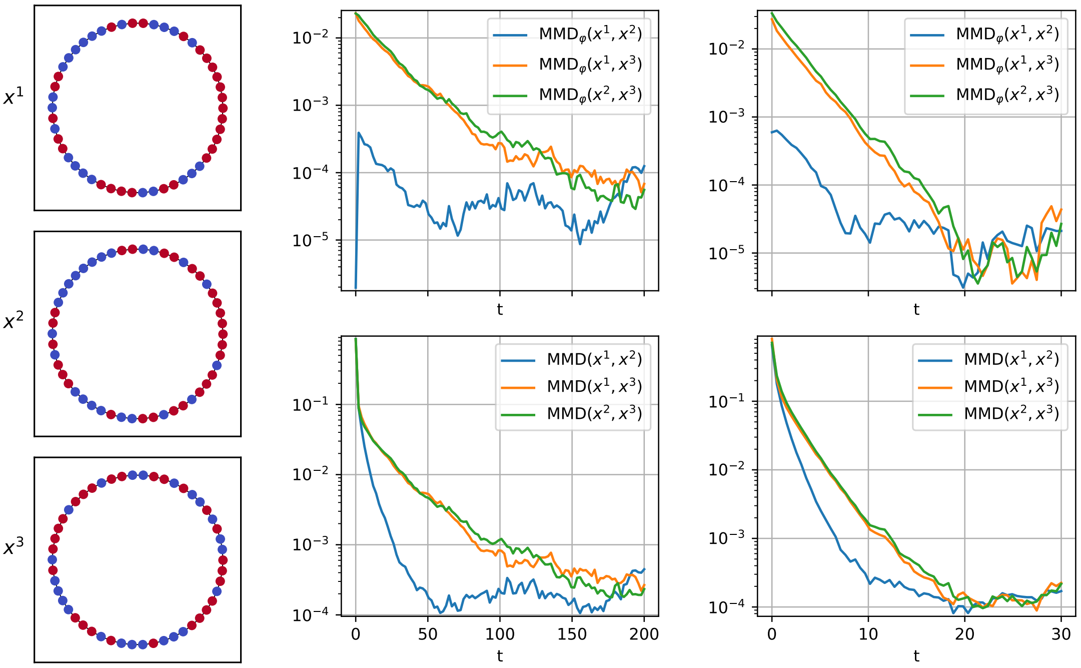
(c) Threshold model.

Figure 6.7.: Validation for the stochastic block model example. For a definition of MMD and MMD_φ see (6.28) and (6.29).

dynamics the time t^* is very small ($t^* \approx 5$) compared to $\hat{t}$. This indicates that the learned CV is good, as discussed above. An interesting difference between the voter model and the threshold model on this network is that the metastable states (where each cluster is dominated by one opinion) persist much longer in the threshold model. As x^1 and x^3 typically move into different metastable states, their MMD stays large much longer in the threshold model.

For the ring network, state x^1 is chosen randomly (i.i.d. for every node), x^2 is chosen such that the collective variable matches, and x^3 is given by a random permutation of node states of x^1, see figure 6.8. Hence, state x^1 and x^3 have the same number of 1's. Here, the time t^* is rather large compared to $\hat{t}$, which shows that this CV is too coarse. This is not surprising because the CV dimension was capped at 5, even though the transition manifold approach suggested a higher dimension. One could improve the quality of this CV (i.e., reduce t^*) at the cost of increasing its dimension and hence complexity.

For the Albert–Barabási network, state x^2 is such that the top 10% of nodes with the largest degrees are in state 1, whereas in x^3 the 10% of nodes with the smallest degrees are in state 1, see figure 6.9. Hence, the total number of 1's is identical, but the CV, which measures the degree-weighted count of 1's, differs substantially. State x^1 is chosen randomly such that it has the same degree-weighted count of 1's as state x^2. This experiment indicates that the learned CV is good because t^* is small compared to $\hat{t}$. Moreover, it is again observed that due to the longer persistence of metastable states in the threshold model, the MMD between x^1 and x^3 stays much larger in the threshold model than in the voter model.

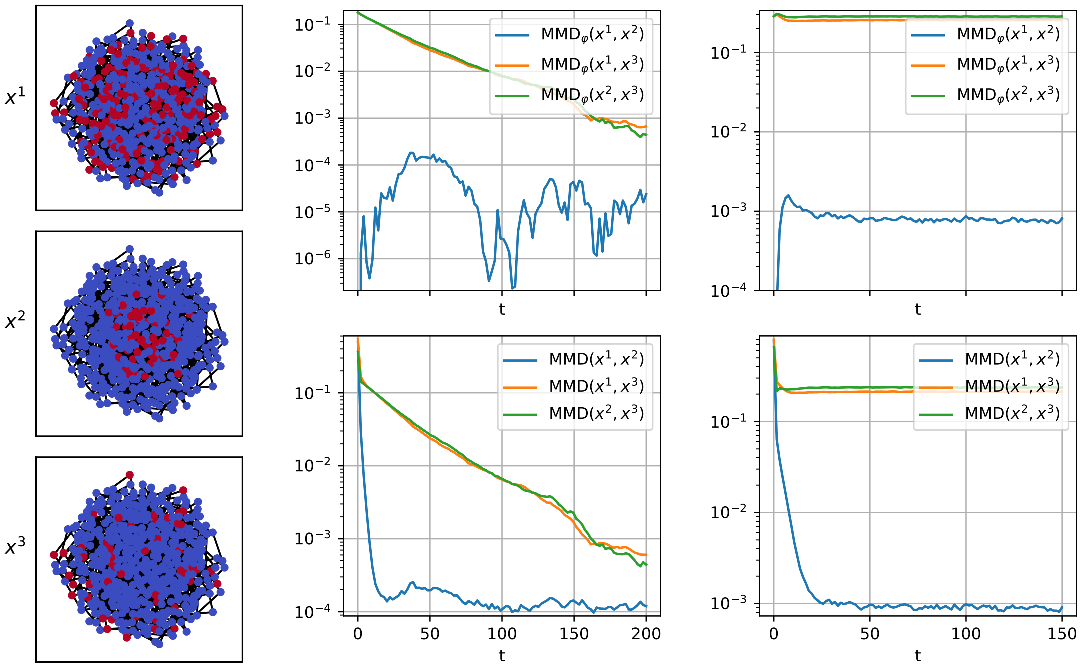


(a) States x^1, x^2, x^3.

(b) Voter model.

(c) Threshold model.

Figure 6.8.: Validation for the ring network example. For a definition of MMD and MMD_φ see (6.28) and (6.29).

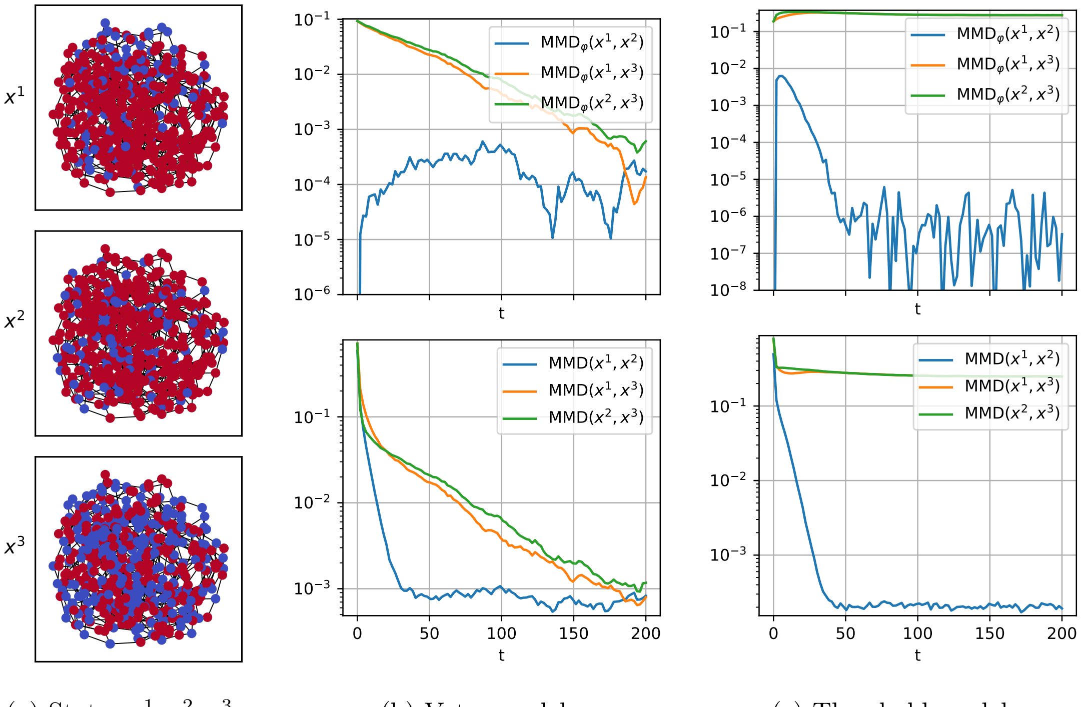


(a) States x^1, x^2, x^3.

(b) Voter model.

(c) Threshold model.

Figure 6.9.: Validation for the Albert–Barabási network example. For a definition of MMD and MMD_φ see (6.28) and (6.29).



(a) States x^1, x^2, x^3. (b) Voter model. (c) Threshold model.

Figure 6.10.: Validation for the random 3-regular network. For a definition of MMD and MMD$_\varphi$ see (6.28) and (6.29).

Finally, figure 6.10 shows that the simple share of 1's is indeed a good CV in the case of the random 3-regular network. States x^1 and x^2 are random permutations with the same number of 1's, whereas x^3 has a different number of 1's. The time t^* is again small compared to $\hat{t}$, which implies that the CV is good.

6.3. Surrogate collective dynamics for the Albert–Barabási model

The standard mean-field equation (2.24), which describes the infinite population dynamics of the continuous-time noisy voter model (CNVM) on sufficiently dense and homogeneous graphs, see section 5.2, is not valid for scale-free networks generated by the Albert–Barabási model. Due to the highly varying node degrees in scale-free networks, the simple state shares, i.e., the percentages of each discrete state in the system, do not contain sufficient information. If for instance the 10% of nodes with the largest degrees are infected in an SI model, the infection spreads much faster than if the 10% of nodes with the smallest degrees would be infected. This intuition was confirmed in the previous section, where the optimal CV that was learned from data actually contained the *degree-weighted* shares. Hence, one could speculate that there exists an equation similar to the mean-field equation (MFE), but applied to the degree-weighted shares, that yields a good approximation for Albert–Barabási networks. The objective of this section is to learn this equation from data and validate it numerically.

Even if the different states, e.g., infected and susceptible, are distributed evenly across nodes of all degrees, the standard MFE does not match the dynamics of the model, as depicted in figure 6.11. Recall the parameter m of the Albert–Barabási model that dictates how many new nodes are added in each iteration of the preferential attachment. Especially for small m the discrepancy between MFE and actual dynamics is large. For larger m on the other hand, the MFE is a much better match. The reason for that is presumably that, as each node has at least m neighbors and the local clustering in Albert–Barabási networks is small, the neighborhood of each node likely represents the global shares quite well if m is large, and hence mean-field theory can be applied.

In the literature, most macroscopic equations for scale-free networks are very high-dimensional as they treat the state shares separately for each node degree occurring in the network [165, 93], i.e., they are heterogeneous approximations (see section 2.4.2). Moreover, they are not exact because in the theoretical derivation some form of stochastic correlation is always ignored, for example the node degree correlation of neighboring nodes, leading to an approximation error. As of now, a low-dimensional equation for the degree-weighted shares can not be found in the literature, neither in the form of a theoretical derivation nor learned from data-driven methods. Due to the difficult nature of the Albert–Barabási model, e.g., because of degree and edge correlations, an analytical derivation is exceptionally hard. Thus, the following data-driven technique will be applied to learn the dynamics of the degree-weighted shares instead.

Sparse identification of nonlinear dynamics (SINDy). In SINDy [166], governing equations of dynamical systems are learned from trajectory data by solving a regression problem using a sparsity-promoting optimizer. Assume that trajectory data from a system $c(t) \in \mathbb{R}^M$ is given, i.e., $c(t_1), \dots, c(t_n)$. The goal is to find a function $f : \mathbb{R}^M \to \mathbb{R}^M$ such that

$$\dot{c}(t) := \frac{\mathrm{d}}{\mathrm{d}t} c(t) = f\big(c(t)\big). \tag{6.30}$$

Let $C \in \mathbb{R}^{n \times M}$, $C_{i,m} := c_m(t_i)$ denote the data matrix and $\dot{C} \in \mathbb{R}^{n \times M}$, $\dot{C}_{i,m} := \dot{c}_m(t_i)$ the matrix of derivatives. When the partial derivatives $\dot{c}_m(t_i)$ are not known, they are

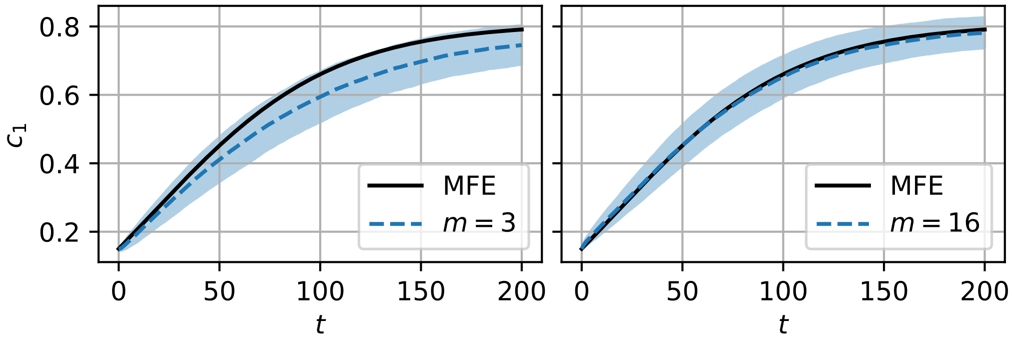


Figure 6.11.: Mean (dashed line) ± standard deviation (shaded area) of the CNVM on Albert–Barabási networks for different preferential attachment parameters m, in comparison with the mean-field solution (MFE).

simply approximated from the trajectory data via finite differences. Furthermore, let a set of library functions $\theta_1, \dots, \theta_\ell$, with $\theta_j : \mathbb{R}^M \to \mathbb{R}$, be given and let the library matrix $\Theta(C) \in \mathbb{R}^{n\times\ell}$ be defined by $\Theta(C)_{i,j} := \theta_j\big(c(t_i)\big)$. Now, the function f is constructed via linear combinations of library functions, i.e., $f(c) = \Theta(c)^T\Xi$ where $\Theta(c) \in \mathbb{R}^\ell$ contains the evaluations of the library functions at c and $\Xi \in \mathbb{R}^{\ell\times M}$ is the matrix of linear coefficients. The optimal coefficients Ξ are then found by solving the regression problem

$$\dot{C} \overset{!}{=} \Theta(C)\,\Xi\,, \tag{6.31}$$

most commonly using a sparse optimizer like the *sequentially thresholded least squares algorithm* (STLSQ) [166] or the *stepwise sparse regressor* (SSR) [167].

Applying SINDy to the CNVM. To acquire the necessary trajectory data, the CNVM with parameters

$$r = \begin{pmatrix} - & 1.01 \\ 0.99 & - \end{pmatrix}, \qquad \tilde{r}_{m,n} = 0.05 \tag{6.32}$$

is simulated on an Albert–Barabási network of size $N = 10^5$ with preferential attachment parameter $m = 3$. From the simulations, trajectories of the degree-weighted share $c(t) \in [0, 1]$ of nodes in state 1 are calculated. These trajectories still exhibit some variance, i.e., they fluctuate slightly around their mean, which would obstruct the learning algorithm. Hence, to make it easier for SINDy, a few trajectories starting in the same initial state are sampled and then averaged to even out the fluctuations. The set of training data then consists of several of these averaged trajectories, starting in different initial states with varying degree-weighted shares. In this example, three averaged trajectories are used as the training data, and a fourth trajectory is sampled to test if the learned equation is valid. The set of monomials up to degree four is employed as the function library and the regression problem is solved using the STLSQ optimizer. Given these inputs, the following equation is obtained by SINDy

$$\frac{\mathrm{d}}{\mathrm{d}t}c(t) = 0.0049 + 0.0063c(t) - 0.0163c(t)^2. \tag{6.33}$$

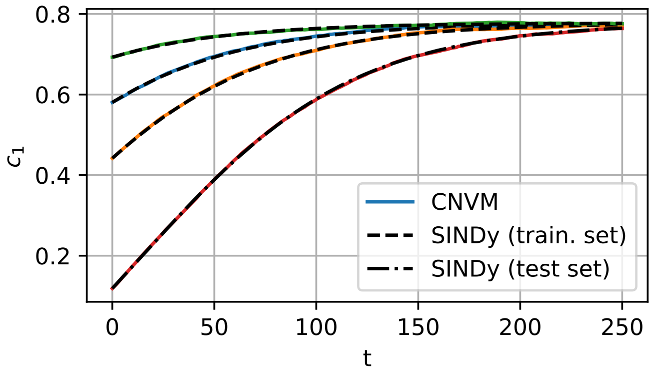


Figure 6.12.: Model simulations of the degree-weighted share c_1, in comparison with the modified mean-field equation (6.33) learned by SINDy.

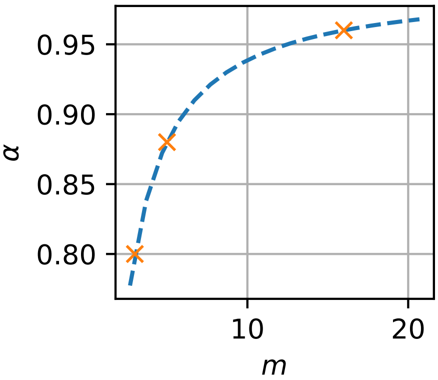


Figure 6.13.: Estimation of the factor α in the modified MFE (6.36) depending on m.

This equation successfully replicates the three training trajectories but also generalizes well to the fourth trajectory with unseen initial state, see figure 6.12.

Note that there are no third and fourth order terms present in the learned equation. In fact, it closely resembles the standard MFE (2.24), which for these parameters is given by

$$\frac{\mathrm{d}}{\mathrm{d}t}c(t) = \tilde{r}_{0,1} + \big(r_{0,1} - r_{1,0} - \tilde{r}_{0,1} - \tilde{r}_{1,0}\big)\, c(t) + \big(r_{1,0} - r_{0,1}\big)\, c(t)^2 \tag{6.34}$$

$$= 0.005 + 0.01\, c(t) - 0.02\, c(t)^2. \tag{6.35}$$

(Recall however that in the context of the standard MFE the shares c are not degree-weighted.) When comparing these two equations, it becomes apparent that the learned equation (6.33) can be obtained from the standard MFE (6.35) by multiplying the rates r (but not $\tilde{r}$) with the factor $\alpha = 0.8$, i.e.,

$$\frac{\mathrm{d}}{\mathrm{d}t}c(t) = \tilde{r}_{0,1} + \big(\alpha\, r_{0,1} - \alpha\, r_{1,0} - \tilde{r}_{0,1} - \tilde{r}_{1,0}\big)\, c(t) + \big(\alpha\, r_{1,0} - \alpha\, r_{0,1}\big)\, c(t)^2. \tag{6.36}$$

Hence, the coefficient of the linear term becomes $(\alpha\, r_{0,1} - \alpha\, r_{1,0} - \tilde{r}_{0,1} - \tilde{r}_{1,0}) = 0.006$ and the coefficient of the quadratic term $(\alpha\, r_{1,0} - \alpha\, r_{0,1}) = -0.016$, which closely matches the learned equation. This suggests that, while the noise remains the same, the infection (or imitation) in the CNVM is effectively about 20% slower on the Albert–Barabási network compared to the standard MFE.

Interestingly, this phenomenon can also be observed for larger values of the preferential attachment parameter m, but the factor α changes based on m. For instance, the same application of SINDy to an Albert–Barabási network with $m = 5$ yields the modified MFE (6.36) with $\alpha = 0.88$, and for $m = 16$ the value $\alpha = 0.96$ is obtained, which agrees well with the earlier observation that the larger m is the more similar the dynamics becomes to the standard MFE. Based on these experiments, an estimation of α in dependence of m is shown in figure 6.13.

It remains to be verified that this modified MFE with rate parameters αr instead of r is not just valid for the specific CNVM parameters used above, but for all choices of r and $\tilde{r}$, or at least a wide range of choices, and for any number of node states M. This would require a substantial amount of thorough testing which is left for future work. In the following only one further example is considered as a verification: the CNVM with $M = 3$ states and parameters

$$r = \begin{pmatrix} - & 8 & 2 \\ 2 & - & 8 \\ 8 & 2 & - \end{pmatrix}, \qquad \tilde{r} = \begin{pmatrix} - & 0.09 & 0.07 \\ 0.07 & - & 0.09 \\ 0.09 & 0.07 & - \end{pmatrix}. \tag{6.37}$$

Figure 6.14 illustrates how accurate the modified MFE is in this example for 9 trajectories started in different initial states. The underlying Albert–Barabási network was constructed with parameter $m = 3$ and $\alpha = 0.8$ was used in the modified MFE as this was the optimal choice in the previous example. It is obvious that the modified MFE produces a much better approximation than the standard MFE and the qualitative behavior matches closely as well. However, there are still some discrepancies with varying severeness depending on the initial state. While these could be finite size effects that vanish as $N \to \infty$ to some extent, it is probable that the modified MFE does not become completely exact, unlike the standard MFE which constitutes a true mean-field limit for certain networks as shown in chapter 5. The small errors that can be seen in figure 6.14 are by no means contradictory to the weighted shares being good CVs, as it is precisely the concept of CVs to facilitate an optimal trade-off between approximation error and dimension reduction. While the errors seem to be quite small, further experiments are necessary to quantify them more thoroughly and to compare them to the errors of other approximations like the pair approximation (see section 2.4.2). It is also possible that, depending on the specific rates r and $\tilde{r}$, other (simple) modifications to the MFE can be performed to further reduce the approximation error. This could also be considered in future work.

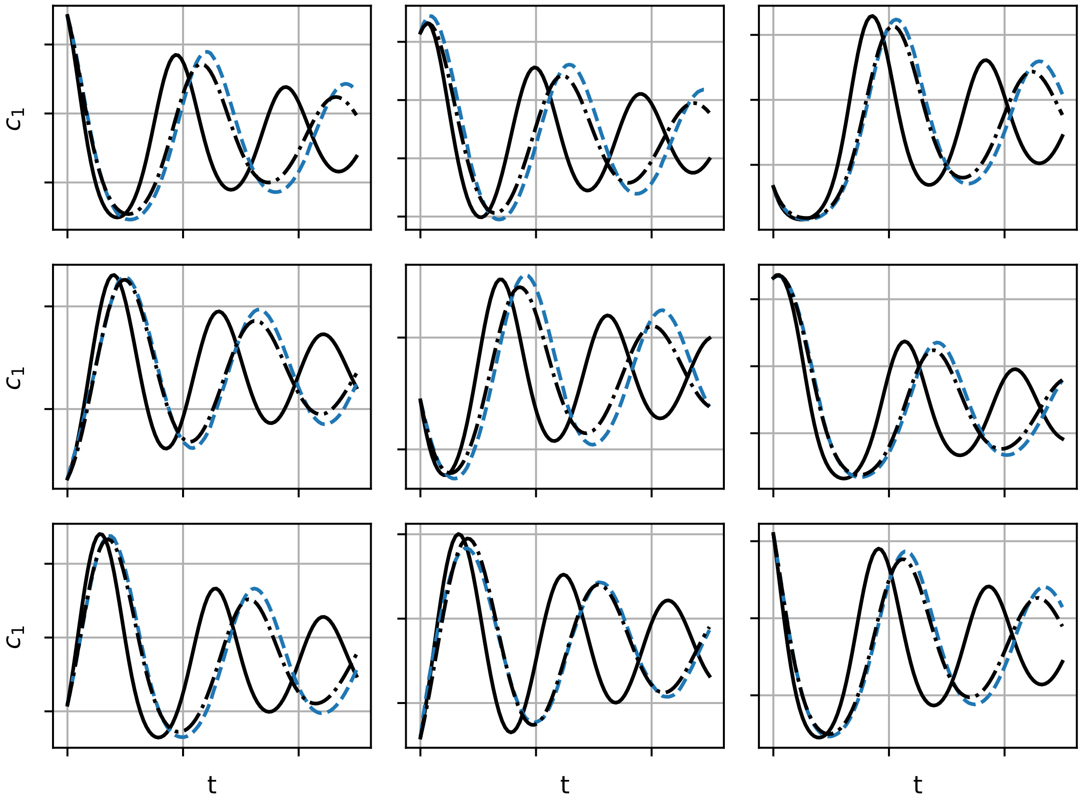


Figure 6.14.: Test of the modified MFE for the CNVM with parameters specified in (6.37) in comparison with model simulations. Blue dotted: CNVM, black solid: standard MFE, black dotted: modified MFE.

7. Ring-shaped networks

This chapter contains several considerations regarding dynamical systems on ring-shaped networks, i.e., the one-dimensional periodic lattice and similar networks. Generally, studying dynamical systems and especially their large population limits on such sparse networks is much harder and requires different approaches than the mean-field theory for dense networks [168]. There are however numerous results for varying dynamical models and underlying graphs in the literature. For ring and lattice structures it is common to derive a large population limit in the form of a PDE resembling the heat equation or diffusion equation that describes the evolution of probability densities in the system [84, 169]. Because the limit dynamics is not necessarily deterministic, extensions to stochastic PDEs have also been studied [82, 83]. It should be noted that these results typically assume a rescaling of time as the population N increases which effectively makes the dynamics run faster. The reason for this rescaling is that, if average path lengths in the sparse network increase rapidly with the number of nodes, the time that information needs to spread through the network also increases. For instance, in an epidemiological model the average time for an infection to spread from a seed node at angle $0°$ on the ring to any other node at angle $\alpha > 0$ typically scales linearly with N, and hence in the limit dynamics on an "infinite" ring the infection does not spread at all. Hence, the rate of infection has to be increased with N, or equivalently time rescaled, to produce a meaningful limit.

Instead of dealing with these types of large population limits, this chapter focuses on approximations for spreading processes on a finite ring and the connection to mean-field theory for dense networks.

First, it is demonstrated in section 7.1 that the mean-field limit of the continuous-time noisy voter model (CNVM) is a poor approximation for the dynamics on the ring, which is not surprising. However, recall that the binomial Watts–Strogatz model (see section 3.1.4) can continuously interpolate between a ring-shaped network and the Erdős–Rényi (ER) model. As the standard mean-field equation (MFE) is the large population limit on ER networks, which was shown in section 5.2.1, but is a poor approximation on the ring, it is interesting to study the approximation error of the MFE in the intermediate *small-world* regime. In this section it becomes apparent that, even if a decent amount of random edges are added to the ring, the approximation error of the MFE is large. This result confirms that different approaches have to be employed to treat ring-shaped networks.

One possible approach to obtain a reduced model is given by moment closure methods, which were introduced in section 2.4.2. Here, in section 7.2, a moment closure method is applied to the CNVM on the ring to obtain a so-called *triplet approximation*. This well-known approach is complemented with a novel data-driven technique to decrease the approximation error by optimizing the closure of equations. The resulting method is more robust than other data-driven algorithms that have to learn the complete dynamics from scratch, and has a lower approximation error than the classical triplet approximation.

Finally, a dense version of the ring graph in which each node is connected to a fixed percentage of its left and right neighbors on the ring is considered in section 7.3. Due to the density, a rescaling of time is not necessary to obtain a meaningful large population limit. Instead, graphon theory (see section 3.2) can be applied to derive a limit PDE describing the evolution of probability densities in the system, which is demonstrated in this section. Furthermore, it is shown that the graphon mean-field limit fails to produce a good approximation if the network becomes too sparse.

7.1. Watts–Strogatz networks and mean-field limits

Recall from section 3.1.4 that the binomial Watts–Strogatz model interpolates between a regular ring lattice (rewiring probability $\beta = 0$), where every node is connected to its nearest $K/2$ left and $K/2$ right neighbors on the ring, and the Erdős–Rényi (ER) model with edge density $p = \frac{K}{N-1}$ (rewiring probability $\beta = 1$). Moreover, in chapter 5 it was shown that the standard mean-field equation (5.82) is the large population limit of the continuous-time noisy voter model (CNVM) on ER graphs if $p \gg \log(N)/N$. Thus, if $K \gg \log N$ in the binomial Watts–Strogatz model, the mean-field equation is the large population limit in the case $\beta = 1$, but for the regular ring ($\beta = 0$) it does not describe the macroscopic dynamics well and is a rather bad approximation. In this section the behavior of the CNVM for intermediate rewiring probabilities $0 < \beta < 1$ is discussed, with a special focus on the approximation quality of the mean-field equation. Recall that, as β increases from 0 to 1, first the average path length becomes small while the clustering coefficient remains large, which is called the *small-world* property. The question whether the mean-field equation constitutes a good approximation in that small-world setting is addressed by the following numerical experiment.

Consider the CNVM with $M = 2$ opinions $\{0, 1\}$ and parameters

$$r = \begin{pmatrix} - & 1.01 \\ 0.99 & - \end{pmatrix}, \qquad \tilde{r}_{m,n} = 0.01. \tag{7.1}$$

The parameter K is chosen such that the mean degree is sufficiently large, $K = \log(N)^2$. Figure 7.1 shows averaged trajectories of the share c_1 of 1's for different rewiring probabilities β of the binomial Watts–Strogatz model on $N = 2500$ nodes. For each experiment, 20% of nodes are in state 1 and the rest in state 0 initially. As expected, the trajectory for $\beta = 1$ is very close to the solution of the mean-field equation (MFE). For smaller β, the trajectories move further away from the MFE and converge to a different steady-state.

Experiments with different network sizes N confirm this behavior, as figure 7.2 illustrates. Interestingly, for all N the approximation error of the MFE is constant or decreases only slightly for small β until it starts to drop rapidly at about $\beta = 10^{-2}$. The error then falls quickly until about $\beta = 10^{-1}$, after which it again remains constant or decreases only slightly. The small-world regime however is in the range $10^{-4} < \beta < 10^{-2}$, where the average path length has already dropped but the average clustering remains large. Hence, the error is still quite large in the small-world regime implying that the MFE is not a suitable approximation for small-world networks.

Moreover, it also becomes apparent from figure 7.2 that the range $10^{-2} < \beta < 10^{-1}$ in which the MFE decreases the most coincides with the range where the average clustering coefficient begins to drop off quickly. Considering the principles of *indistinguishability* and *interchangeability* in mean-field theory (see section 2.4.1), it is not surprising that the

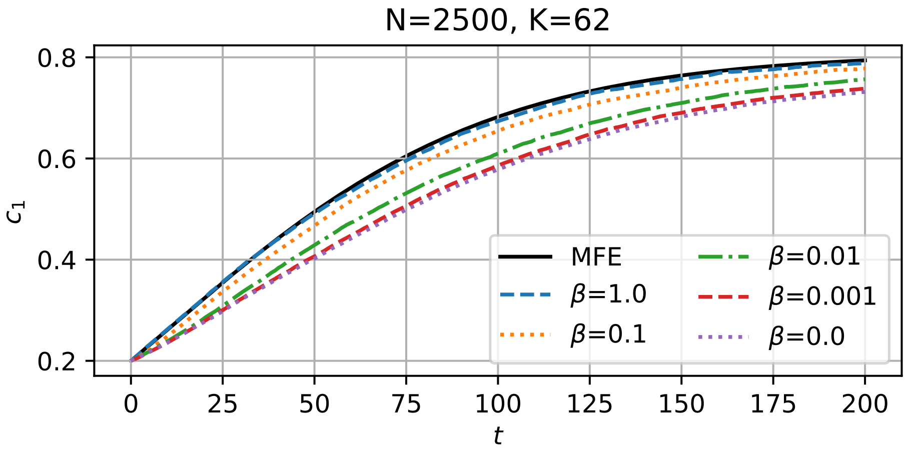


Figure 7.1.: CNVM on Watts–Strogatz networks of varying rewiring probability β, in comparison with the mean-field equation (MFE). The plot shows the share c_1, averaged over 1000 simulations.

error of the MFE only starts to decrease when local structures become less prevalent in the network, which is detected by the falling clustering coefficient.

While this numerical experiment clearly indicates that the MFE is not a good approximation for the ring or for small-world networks, it is an open question whether the MFE is an exact large population limit only for rewiring probability $\beta = 1$ or if there is a phase transition at some critical value β_c such that the MFE is valid for all $\beta > \beta_c$. Since $\beta < 1$ implies that some local structure of the ring remains, it seems more likely that the MFE is only an exact limit for $\beta = 1$ and merely an approximation for $\beta < 1$ with an error that is smaller the closer β is to 1. Further investigation is required to answer this question definitely.

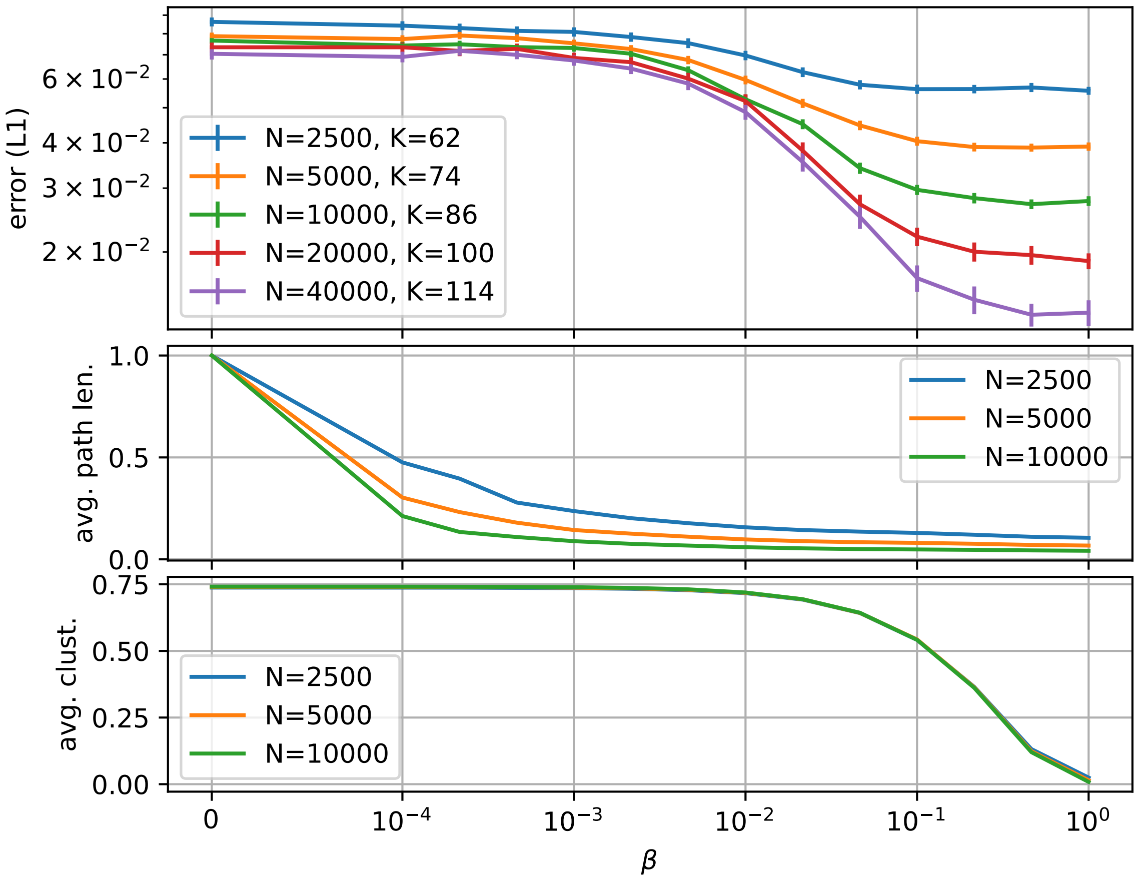


Figure 7.2.: The L_1-error between mean-field equation and mean of CNVM (top), the normalized average path length (center), and the average cluster coefficient (bottom), for binomial Watts–Strogatz networks of varying rewiring probability β and varying size N.

7.2. Learning a moment closure

This section is concerned with finding a low-dimensional approximation to the macroscopic behavior of the continuous-time noisy voter model (CNVM) on a standard ring network. A triplet approximation (see section 2.4.2) is a natural choice for this setting but requires a closure of the evolution equations that introduces an error. First, in section 7.2.1 a classical approach is presented for which the closure is based on assumptions of stochastic independence. While this classical triplet approximation generally works quite well, it can show a large approximation error in some cases. "Off-the-shelf" algorithms for learning dynamics from data, like *SINDy*, can be used to learn better evolution equations, which is demonstrated in section 7.2.2. However, since these general purpose algorithms are very flexible, it is often hard to find optimal hyperparameters for a specific setting, and the learned dynamics may show behavior that is unwanted or unrealistic for a certain system. For example, in this specific case of the CNVM on a ring, SINDy often learns dynamics where the sum of the shares of nodes in each state can not only exceed 1, but even explodes to infinity. A novel and more robust strategy for improving the approximation error is to only apply a data-driven learning algorithm to the part of the triplet approximation that introduces an error: the closure. This method of only learning the closure is presented in section 7.2.3.

The contents of this section are strongly inspired by the author's publication [3].

7.2.1. A classical triplet approximation

Consider again the continuous-time noisy voter model (CNVM), see section 2.2 for details. It is assumed that nodes have one of $M = 2$ states, $x_i \in \{0, 1\}$. For a standard ring network with N nodes, in which each node is connected to one left and one right neighbor, a triplet approximation is a natural choice. Given a state $x \in \{0, 1\}^N$, define the collective variables $C(x) := (C_0^0(x), C_0^1(x), C_0^2(x), C_1^0(x), C_1^1(x), C_1^2(x))$, where $C_m^k(x)$ denotes the share of nodes that have state m and have k neighbors of the other state $n = 1 - m$. For abbreviation the reduced state will be denoted as $C(x) = c$. Note that $c_m^0 + c_m^1 + c_m^2 =: c_m$ is the total share of opinion m and hence $c_0 + c_1 = \|c\|_1 = 1$. Inspecting the propensity functions for the CNVM on the ring yields

$$\alpha_{m\to n}(x) := \sum_{i:x_i=m} \left(r_{m,n} \frac{d_{i,n}(x)}{d_i} + \tilde{r}_{m,n} \right) \tag{7.2}$$

$$= \frac{r_{m,n}}{2} \sum_{i:x_i=m} d_{i,n}(x) + N c_m \tilde{r}_{m,n} \tag{7.3}$$

$$= N \left(\frac{r_{m,n}}{2} (c_m^1 + 2c_m^2) + c_m \tilde{r}_{m,n} \right). \tag{7.4}$$

Thus, a reduced propensity function $\tilde{\alpha}_{m\to n}(c) = N^{-1} \alpha_{m\to n}(x)$ for the collective variable $c = C(x)$ that counts triplets occurs naturally due to the special structure of the ring. However, in order to derive an evolution equation for c, it is necessary to specify the state-change vectors for the possible transitions, which is where the moment closure becomes necessary.

The difficulty of deriving the state-change vectors, i.e., the impact of the state transitions on c, and their associated rates is illustrated best using an example. Consider the case that a node i transitions from opinion m to n and has $k = 0$ neighbors of opinion n. This transition occurs at the rate $\tilde{r}_{m,n}$, purely due to noise. It is clear that before the transition it is $(x_{i-1}, x_i, x_{i+1}) = (m, m, m)$ and that because of the transition of node i the share c^0_m decreases by $1/N$ and the share c^2_n increases by $1/N$. Additionally, the neighborhoods of node $(i-1)$ and node $(i+1)$ change as well, but it is not possible to say how this impacts c because that depends on the unknown states of nodes $(i-2)$ and $(i+2)$. If for instance $(x_{i-2}, x_{i+2}) = (m, m)$, then nodes $(i-1)$ and $(i+1)$ would count towards c^0_m before and towards c^1_m after the transition of node i, which would lead to the total state-change vector $N^{-1}(-3, 2, 0, 0, 0, 1)$. The other three possible scenarios $(x_{i-2}, x_{i+2}) = (m, n), (n, m), (n, n)$ lead to other potentially different state-change vectors. The question is therefore how to distribute the total rate $\tilde{r}_{m,n}$ of this transition to the four possible scenarios.

A standard theme in moment closure methods is to close equations by assuming stochastic independence of certain quantities. For instance, in the example above it could be assumed that the state of node $(i-2)$ is independent of the other nodes and is simply randomly distributed according to the relative frequencies dictated by c. Hence, node $(i-2)$ is assumed to have state m with probability $c^0_m/(c^0_m + 0.5c^1_m)$. The same reasoning applied to node $(i+2)$ yields a probability of the scenario $(x_{i-2}, x_{i+2}) = (m, m)$ of $\big(c^0_m/(c^0_m + 0.5c^1_m)\big)^2$, which will be the percentage of the total rate $\tilde{r}_{m,n}$ that is assigned to the associated state-change vector.

Following this procedure for all possible cases, i.e., node i having opinion $m = 0, 1$ and having $k = 0, 1, 2$ neighbors of opinion n, and nodes $(i-2)$ and $(i+2)$ having opinions $(x_{i-2}, x_{i+2}) = (m, m), (m, n), (n, m), (n, n)$ yields a total of $2 \cdot 3 \cdot 4 = 24$ state-change vectors v_j with associated rates $\tilde{\alpha}_j(c)$. The rates are given by multiplying a base rate for that transition, which would be $c^0_m \tilde{r}_{m,n}$ in the above example, with a probability that is assigned to the scenario j by assuming stochastic independence of certain node states. Due to the symmetry of the cases $(x_{i-2}, x_{i+2}) = (m, n)$ and $(x_{i-2}, x_{i+2}) = (n, m)$ if $k = 0$ or $k = 2$, the total number of cases can be reduced from 24 to 20. An overview is presented in table 7.1. Finally, combining everything yields the evolution equation

$$\frac{\mathrm{d}}{\mathrm{d}t} c(t) = \sum_{j=1}^{20} \tilde{\alpha}_j\big(c(t)\big) v_j. \tag{7.5}$$

Although the collective variable c is 6-dimensional, there are actually only 4 degrees of freedom due to the periodicity of the ring introducing certain symmetries. Consider any fixed system state on the ring and define the random variable $\boldsymbol{X} = (\boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{X}_3)$ such that $\mathbb{P}(\boldsymbol{X} = (x_1, x_2, x_3))$ is the probability of seeing (x_1, x_2, x_3) when randomly picking 3 consecutive nodes from the ring. Then it is easy to see that $\mathbb{P}(\boldsymbol{X} = (0, 0, 0)) = c^0_0$ and $\mathbb{P}(\boldsymbol{X} \in \{(0, 0, 1), (1, 0, 0)\}) = c^1_0$. Note that, due to periodicity, the number of occurrences of $(0, 0, 1)$ and $(1, 0, 0)$ are always exactly equal. Thus, it follows that

$$\mathbb{P}(\boldsymbol{X} = (0, 0, 1)) = \mathbb{P}(\boldsymbol{X} = (1, 0, 0)) = \frac{1}{2} c^1_0, \tag{7.6}$$

and with the same argumentation

$$\mathbb{P}(\boldsymbol{X} = (0, 1, 1)) = \mathbb{P}(\boldsymbol{X} = (1, 1, 0)) = \frac{1}{2} c^1_1. \tag{7.7}$$

Table 7.1.: The 20 state-change vectors and associated rates for the triplet approximation can be obtained from this table by setting $m = 0, n = 1$ or $m = 1, n = 0$. The state-change vectors are written in the format $(c_m^0, c_m^1, c_m^2, c_n^0, c_n^1, c_n^2)$. For $k = 0$ and $k = 2$, two of the four cases can be merged into one due to symmetry; hence there are only three entries.

k	$(x_{i-2}, \ldots, x_{i+2})$	state-change vector v_j	rate $\tilde{\alpha}_j(c)$
0	(m, m, m, m, m)	$(-3, 2, 0, 0, 0, 1)$	$c_m^0 \tilde{r}_{m,n} \cdot \left(\frac{c_m^0}{c_m^0+0.5c_m^1}\right)^2$
0	(m, m, m, m, n)	$(-2, 0, 1, 0, 0, 1)$	$c_m^0 \tilde{r}_{m,n} \cdot 2\frac{c_m^0\, 0.5c_m^1}{(c_m^0+0.5c_m^1)^2}$
0	(n, m, m, m, n)	$(-1, -2, 2, 0, 0, 1)$	$c_m^0 \tilde{r}_{m,n} \cdot \left(\frac{0.5c_m^1}{c_m^0+0.5c_m^1}\right)^2$
1	(m, m, m, n, m)	$(-1, 0, 0, 0, 2, -1)$	$c_m^1 (0.5\, r_{m,n} + \tilde{r}_{m,n}) \cdot \frac{c_m^0}{c_m^0+0.5c_m^1} \frac{c_n^2}{c_n^2+0.5c_n^1}$
1	(m, m, m, n, n)	$(-1, 0, 0, 1, 0, 0)$	$c_m^1 (0.5\, r_{m,n} + \tilde{r}_{m,n}) \cdot \frac{c_m^0}{c_m^0+0.5c_m^1} \frac{0.5c_n^1}{c_n^2+0.5c_n^1}$
1	(n, m, m, n, m)	$(0, -2, 1, 0, 2, -1)$	$c_m^1 (0.5\, r_{m,n} + \tilde{r}_{m,n}) \cdot \frac{0.5c_m^1}{c_m^0+0.5c_m^1} \frac{c_n^2}{c_n^2+0.5c_n^1}$
1	(n, m, m, n, n)	$(0, -2, 1, 1, 0, 0)$	$c_m^1 (0.5\, r_{m,n} + \tilde{r}_{m,n}) \cdot \frac{0.5c_m^1}{c_m^0+0.5c_m^1} \frac{0.5c_n^1}{c_n^2+0.5c_n^1}$
2	(m, n, m, n, m)	$(0, 0, -1, 1, 2, -2)$	$c_m^2 (r_{m,n} + \tilde{r}_{m,n}) \cdot \left(\frac{c_n^2}{c_n^2+0.5c_n^1}\right)^2$
2	(n, n, m, n, m)	$(0, 0, -1, 2, 0, -1)$	$c_m^2 (r_{m,n} + \tilde{r}_{m,n}) \cdot 2\frac{c_n^2\, 0.5c_n^1}{c_n^2+0.5c_n^1}$
2	(n, n, m, n, n)	$(0, 0, -1, 3, -2, 0)$	$c_m^2 (r_{m,n} + \tilde{r}_{m,n}) \cdot \left(\frac{0.5c_n^1}{c_n^2+0.5c_n^1}\right)^2$

As a consequence, it is

$$\mathbb{P}((\boldsymbol{X_1}, \boldsymbol{X_2}) = (0,1)) = \mathbb{P}(\boldsymbol{X} = (0,1,0)) + \mathbb{P}(\boldsymbol{X} = (0,1,1)) = c_1^2 + \frac{1}{2}c_1^1 \tag{7.8}$$

and

$$\mathbb{P}((\boldsymbol{X_1}, \boldsymbol{X_2}) = (1,0)) = \mathbb{P}(\boldsymbol{X} = (1,0,0)) + \mathbb{P}(\boldsymbol{X} = (1,0,1)) = \frac{1}{2}c_0^1 + c_0^2. \tag{7.9}$$

But again, due to the periodicity, the number of occurrences of $(0,1)$ and $(1,0)$ are exactly equal, which implies

$$c_0^2 + \frac{1}{2}c_0^1 = c_1^2 + \frac{1}{2}c_1^1. \tag{7.10}$$

Thus, the triplet approximation derived above is effectively 4-dimensional. For example, if $(c_0^0, c_0^1, c_0^2, c_1^1)$ are given, then c_1^2 is determined by (7.10), and then c_1^0 by $\|c\|_1 = 1$.

Numerical example. Consider the ring with $N = 2000$ nodes and the CNVM with parameters

$$r = \begin{pmatrix} - & 1.01 \\ 0.99 & - \end{pmatrix}, \qquad \tilde{r}_{m,n} = 0.01. \tag{7.11}$$

Figure 7.3 shows the triplet approximation (7.5) in comparison with the actual model for two different initial conditions. While the qualitative behavior including the steady state

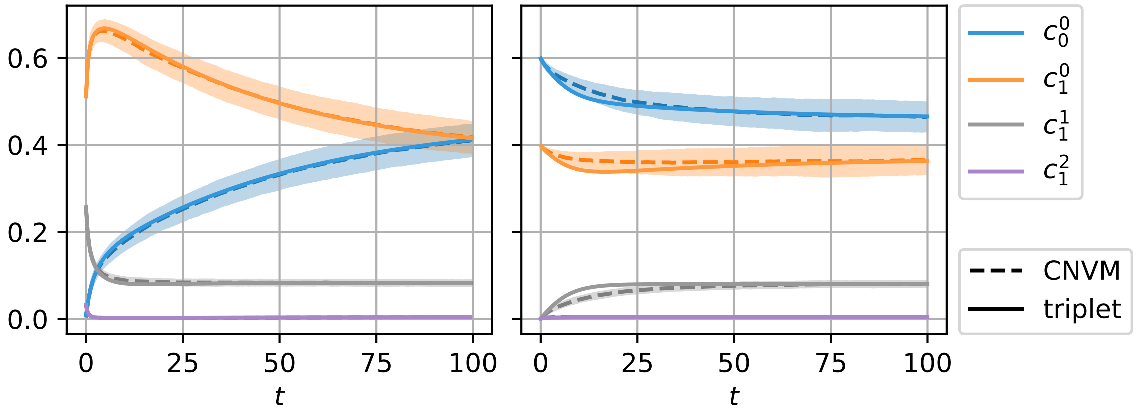


Figure 7.3.: Mean (dashed line) ± standard deviation (shaded area) of the CNVM on the ring with $N = 2000$ nodes, in comparison with the triplet approximation (7.5), for two different initial states.

matches very well, there is a slight approximation error in the transient phase. As the figure illustrates, the magnitude of this error may differ substantially depending on the initial state. This implies that the assumption of a certain stochastic independence, which was used to close the equations, is clearly not always satisfied to a sufficient extent.

An interesting characteristic of the CNVM on a ring is that it is reminiscent of a fast-slow dynamical system [170], i.e., in this example the system quickly moves to a *slow manifold* that satisfies $c_1^1 \approx 0.09$ and $c_1^2 \approx 0.005$. After that initial relaxation only c_0^0 and c_1^0 continue to evolve, but at a slower timescale. The reason for this behavior of the CNVM is that state configurations $(x_{i-1}, x_i, x_{i+1}) = (n, m, n)$ are dynamically short-lived, i.e., the central node will very likely switch to n resulting in the more stable configuration (n, n, n). As a result, in the initial phase large sections where all nodes have the same state will form on the ring. Although one of these sections can be broken up into two due to noise, and two sections can merge to form one larger section, the overall number of sections does not change much over time, as can be seen by the quick stabilization of c_1^1. After the formation of these consecutive sections in the initial phase, the dynamics is largely characterized by neighboring sections of different opinion moving their border by “infecting” the next node behind the interface, leading to an increase or decrease of c_0^0 and c_1^0.

7.2.2. Learning an approximation with SINDy

In this section, the popular technique called *SINDy* (Sparse Identification of Nonlinear Dynamics) [166] is applied to learn the dynamics of the CV c from data. The aim is to discover a better approximation than the triplet approximation that was derived in the previous section.

Recall from section 6.3 that SINDy works as follows: Given a time-series of trajectory data and a library of basis functions, the derivatives at the trajectory points are approximated numerically (e.g., via finite differences), and then an optimal linear combination of basis functions is calculated to approximate the derivatives (i.e., the dynamics), typically via regression including a sparsity promoting regularization. Hence, there are

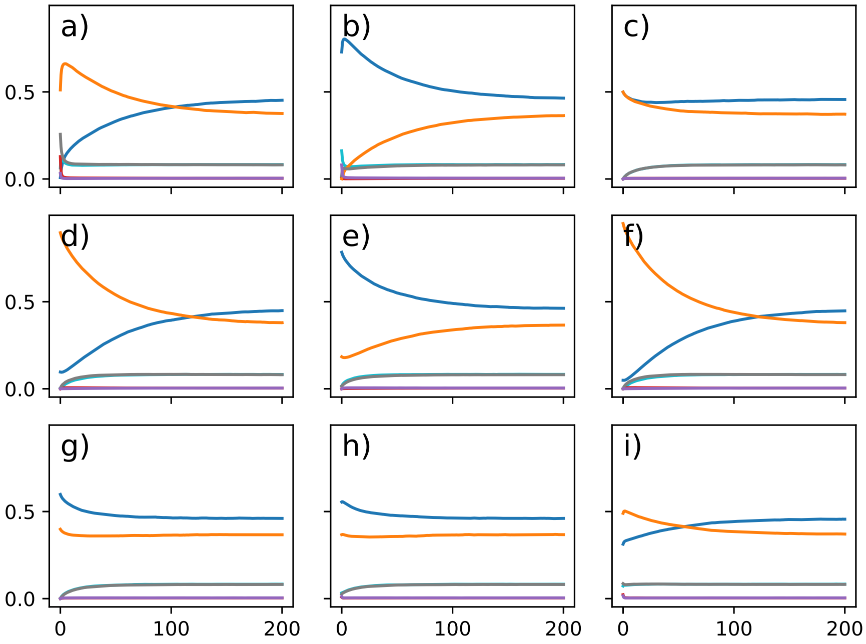


Figure 7.4.: The training data consists of 9 trajectories of the CVs $\boldsymbol{c}(t)$, starting from different initial conditions and averaged over an ensemble of 500 simulations each. The trajectories are labeled from a) to i).

many hyperparameters to be considered when applying SINDy: the method to generate derivatives, the library of basis functions, and the optimizer and optimizer parameters (especially regarding the desired sparsity of coefficients).

In this experiment, the data set consists of 9 trajectories of c with different initial conditions, obtained by averaging over an ensemble of CNVM simulations, see figure 7.4. In order to find optimal hyperparameters, a grid search is combined with leave-one-out cross-validation, i.e., for every choice of hyperparameters 9 models are trained, so that model i has access to all except the i-th trajectory, and the error associated with these hyperparameters is measured as the deviation from the data in infinity norm[1] on the validation trajectory i, averaged over all 9 models. This procedure showed that SINDy is rather sensitive to the choice of hyperparameters. For many choices the dynamics produced by SINDy is vastly unstable such that the system state has to be artificially bounded to not produce an extremely large error. The best choice of hyperparameters was found using central finite differences for calculating the derivatives, polynomials of degree less than 2 as basis functions, and the SSR optimizer [167]. The dynamics produced by SINDy and its error is visualized in figure 7.5. Note that the average error is significantly lower than the error of the triplet approximation from the previous section. However, for some trajectories the dynamics learned by SINDy has a bigger error than the triplet approximation. As discovering the macroscopic dynamics using SINDy was not robust and required a lot of trial and error in selecting working hyperparameters, a more informed learning approach will be discussed in the next section.

[1] over time and over the components of c

7.2.3. Learning only the closure

In the derivation of the triplet approximation in section 7.2.1 an error was introduced by distributing a transition rate among four associated scenarios under independency assumptions that are not correct. In this section, a data-driven method is presented to learn a better way of distributing the transition rates. While the SINDy technique from the previous section has to learn the complete dynamics without prior knowledge, the method presented here targets precisely the unknown part in the triplet approximation, thus learning an optimal closure of the equations.

In the triplet approximation in section 7.2.1, the distribution of transition rates is dictated by a vector $\sigma(c) \in \mathbb{R}^{20}$ that can be derived based on certain independency assumptions. The first three entries of $\sigma(c)$ are associated to the case $(x_{i-1}, x_i, x_{i+1}) = (0,0,0)$ and describe how the transition rate $c_0^0 \tilde{r}_{0,1}$ should be distributed among the scenarios of different node states (x_{i-2}, x_{i+2}). For example, the case $(0,0,0,0,0)$ receives a share of $\sigma(c)_1 = (c_0^0/(c_0^0 + 0.5c_0^1))^2$, and the case $\{(0,0,0,0,1),(1,0,0,0,0)\}$ a share of $\sigma(c)_2 = 2 \cdot c_0^0 0.5 c_0^1/(c_0^0 + 0.5c_0^1)^2$, see table 7.1. The aim is to learn a better function $c \mapsto \sigma(c)$ to specify these shares of transition rates, which would result in a closed evolution equation for c with a smaller approximation error. To obtain the required training data, the occurrences of all possible 5-tuples of consecutive node states $(x_{i-2}, \dots, x_{i+2}) \in \{0,1\}^5$ are counted in every snapshot of the training trajectories, which enables the calculation of the shares of the transition rates. Hence, for each snapshot $x(t_i) \in \{0,1\}^N$ the CVs $c(t_i) \in \mathbb{R}^6$ and the relative frequencies $\sigma(c(t_i)) \in \mathbb{R}^{20}$ of the 20 cases (see table 7.1) are calculated. For example, the first three entries of $\sigma(c(t_i))$ describe how often the three cases $(0,0,0,0,0)$, $\{(0,0,0,0,1),(1,0,0,0,0)\}$, and $(1,0,0,0,1)$ occur, given that the central three nodes are $(0,0,0)$. Thus, these three entries sum to 1 and will be used as the shares when distributing the total transition rate $c_0^0 \tilde{r}_{0,1}$ among the three cases.

To learn the function $c \mapsto \sigma(c)$, an L_1-regularized linear regression is applied. Given a library of functions $\theta_1, \dots, \theta_\ell$, $\theta_j : \mathbb{R}^6 \to \mathbb{R}$, define again the data library matrix $\Theta(C) \in \mathbb{R}^{n \times \ell}$ by $\Theta(C)_{i,j} := \theta_j\big(c(t_i)\big)$. Moreover, store the frequencies of the 20 cases for each snapshot in a matrix $\Sigma \in \mathbb{R}^{n \times 20}$, $\Sigma_{i,:} := \sigma(c(t_i))$. Then optimal sparse coefficients $\Xi \in \mathbb{R}^{\ell \times 20}$ with

$$\Theta(C)\,\Xi \approx \Sigma \tag{7.12}$$

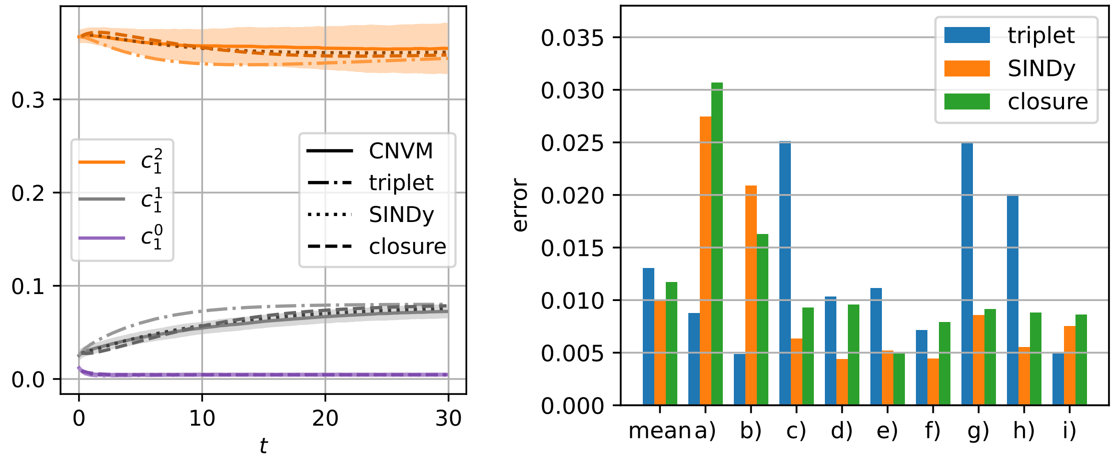
are calculated via the LASSO method [158]. The learned distribution $\sigma(c) = \Theta(c)^T \Xi$ of the transition rates is then employed to close the evolution equations.

A grid search combined with leave-one-out cross-validation is again applied to find optimal hyperparameters. In this example, the best results have been obtained using a library of polynomials with degree smaller than four. As hoped for, this approach of learning only the closure is significantly more robust than SINDy as it never produced unstable dynamics, see table 7.2. However, the error of this method is slightly larger than the error of SINDy, but still lower than the standard triplet approximation, see figure 7.5. This is somewhat expected as this method has a more restricted search space compared to SINDy learning the complete dynamics. Note also that it is not inferior to SINDy on all 9 trajectories, e.g., it is superior to SINDy on trajectory b).

Further testing also showed that this approach of learning the closure requires significantly less data than SINDy to produce good results, i.e., it is possible to outperform the standard triplet approximation using only 5 of the 9 training trajectories while SINDy needs at least 7.

optimizer	library of polynomials of degree		
	1	2	3
STLSQ	33% | 0%	44% | 0%	67% | 0%
SSR	0% | 0%	67% | 0%	100% | 0%

Table 7.2.: Instability of SINDy | closure. The entries show for how many of the 9 training trajectories (see figure 7.4) the learned dynamics is not stable. The depicted hyperparameters (optimizer and library) are only a subselection of all tested hyperparameters.



(a) Approximation of trajectory h).

(b) The error for the 9 trajectories is measured as maximum deviation from data, i.e., in infinity norm.

Figure 7.5.: Comparison of triplet approximation (section 7.2.1), SINDy (section 7.2.2), and triplet approximation with learned closure (section 7.2.3).

7.3. A graphon approximation

In section 7.1 it was discussed that the standard mean-field equation (5.82) is a poor approximation of the continuous-time noisy voter model (CNVM) on a ring graph where each node is connected to its $K/2$ nearest left and $K/2$ nearest right neighbors on the ring. If the node degree K is $\mathcal{O}(N)$, i.e., each node is connected to a non-vanishing percentage of all nodes, the resulting graphs are dense and converge to a graphon in the graph limit, see section 3.2 for more details on graphons. This allows the formulation of a mean-field limit in the form of a partial differential equation (PDE) that describes the evolution of the state probability distribution at each location on the ring. While this PDE is an exact large population limit (not an approximation), it is reliant on the density of the graphs and fails to provide a good approximation for finite networks if K is chosen too small compared to N, which will be demonstrated in this section.

The graphon mean-field limit. In reference [85], which provides a proof for the convergence to a graphon mean-field limit for certain systems, a slightly different definition of the transition rate matrix $Q_i \in \mathbb{R}^{M\times M}$ of node i is employed. Instead of taking the system state x as an input like previously in this thesis, the rate matrix is based on the *neighborhood vector*

$$\phi_i := \frac{1}{N}\big(d_{i,1}, \dots, d_{i,M}\big)^T \in \mathbb{R}^M, \tag{7.13}$$

where as before $d_{i,m}$ is the number of neighbors of node i that have state $m \in [M]$. Thus, the transition rates of node i can only depend on the number of each state in the direct neighborhood of i, which is more restrictive than the definition here in chapter 5 that allows dependence on the whole state x. However, many popular models fit into this framework. Another restriction is that each node is assumed to have the same transition rate function $Q_i = Q$ for all i. Then, if a system of this type is considered on a sequence of graphs that converges to a graphon $W : \mathbb{R}^2 \to \mathbb{R}$, the mean field limit is given by the integro-PDE

$$\partial_t u(t,x) = Q\big(\mathbb{W}u(t,x)\big)^T u(t,x), \tag{7.14}$$

where $u(t,x) \in [0,1]^M$, $\|u(t,x)\|_1 = 1$, describes the probability distribution of the state at location $x \in [0,1]$ on the ring, and the *graphon operator* $\mathbb{W}$ is defined by

$$(\mathbb{W}f)(x) = \int_0^1 W(x,y) f(y) \mathrm{d}y. \tag{7.15}$$

Application to the ring. To achieve sufficient density, it is assumed that each node is connected to its εN left and εN right neighbors on the ring for a fixed $\varepsilon \in (0, \frac{1}{2})$, i.e., $K = 2\varepsilon N$. The resulting graphon

$$W(x,y) = \begin{cases} 1, & \min\big(|x-y|, 1-|x-y|\big) \le \varepsilon \\ 0, & \text{else} \end{cases} \tag{7.16}$$

is 1 on a diagonal band of width 2ε. Since every node has the same degree $2\varepsilon N$, the transition rate matrix of the CNVM (with parameter $\alpha = 0$, see (2.8)) based on the neighborhood vector is given by

$$Q(\phi_i)_{m,n} := r_{m,n}\frac{\phi_{i,m}}{2\varepsilon} + \tilde{r}_{m,n} \tag{7.17}$$

$$= r_{m,n}\frac{d_{i,n}}{2\varepsilon N} + \tilde{r}_{m,n}. \tag{7.18}$$

Inserting everything into (7.14) yields the mean-field equation.

Numerical example. Consider an SIS model with infection rate $\beta > 0$ and recovery rate 1, which is a special case of the CNVM with $M = 2$ states denoted by $\{S, I\}$. The associated transition rate matrix is given by

$$Q(\phi_i) = \begin{pmatrix} -\beta\frac{\phi_{i,I}}{2\varepsilon} & \beta\frac{\phi_{i,I}}{2\varepsilon} \\ 1 & -1 \end{pmatrix}. \tag{7.19}$$

Because $M = 2$, it is sufficient to consider the one-dimensional variable $u(t, x)$ describing the probability of being infected, i.e., in state I. Inserting into (7.14) yields the mean-field limit

$$\partial_t u(t,x) = \beta\frac{1}{2\varepsilon}\mathbb{W}u(t,x)\,\big(1 - u(t,x)\big) - u(t,x). \tag{7.20}$$

A finite-element approach will be utilized to solve this PDE numerically. The spatial domain $[0, 1]$ is discretized into N intervals of length $1/N$ and a piecewise-constant solution on these intervals is assumed. Let $\hat{u}_i(t) \in [0, 1]$ denote the value on the i-th interval. This also has the interesting interpretation that, when considering the finite-size ring graph with N nodes and $K = 2\varepsilon N$, the value $\hat{u}_i(t)$ approximates the probability that node i is infected. The associated adjacency matrix A can be used to define the discretization $\mathbb{W}_N$ of the graphon operator

$$(\mathbb{W}_N v)_i := \sum_{j=1}^{N} A_{i,j} v_j \frac{1}{N}, \tag{7.21}$$

where $v \in \mathbb{R}^N$. Thus, the discretized mean-field limit is given by the system of N coupled ODEs

$$\frac{\mathrm{d}}{\mathrm{d}t}\hat{u}_i(t) = -\hat{u}_i(t) + \frac{\beta}{2\varepsilon}(1 - \hat{u}_i(t))(\mathbb{W}_N\hat{u}(t))_i \tag{7.22}$$

$$= -\hat{u}_i(t) + \frac{\beta}{2\varepsilon}(1 - \hat{u}_i(t))\sum_j A_{i,j}\hat{u}_j(t)\frac{1}{N} \tag{7.23}$$

for $i = 1, \ldots, N$.

Figure 7.6 illustrates the approximation error of the mean-field limit (7.20) for a ring graph with $N = 1000$ nodes. The correct probability $\hat{u}_i(t)$ that node i is infected at time t is estimated from 1000 model simulations. The initial conditions are such that 100 consecutive nodes are infected and the rest are susceptible. It is apparent from the figure that the mean-field limit only provides a reasonable approximation if ε is large enough in relation to N. While the mean-field limit approximates the correct dynamics well for $\varepsilon = 0.1$ (every node has degree $K = 200$), the error is already quite large for $\varepsilon = 0.05$. For $\varepsilon = 0.01$ (every node has degree $K = 20$), the evolution predicted by the mean-field

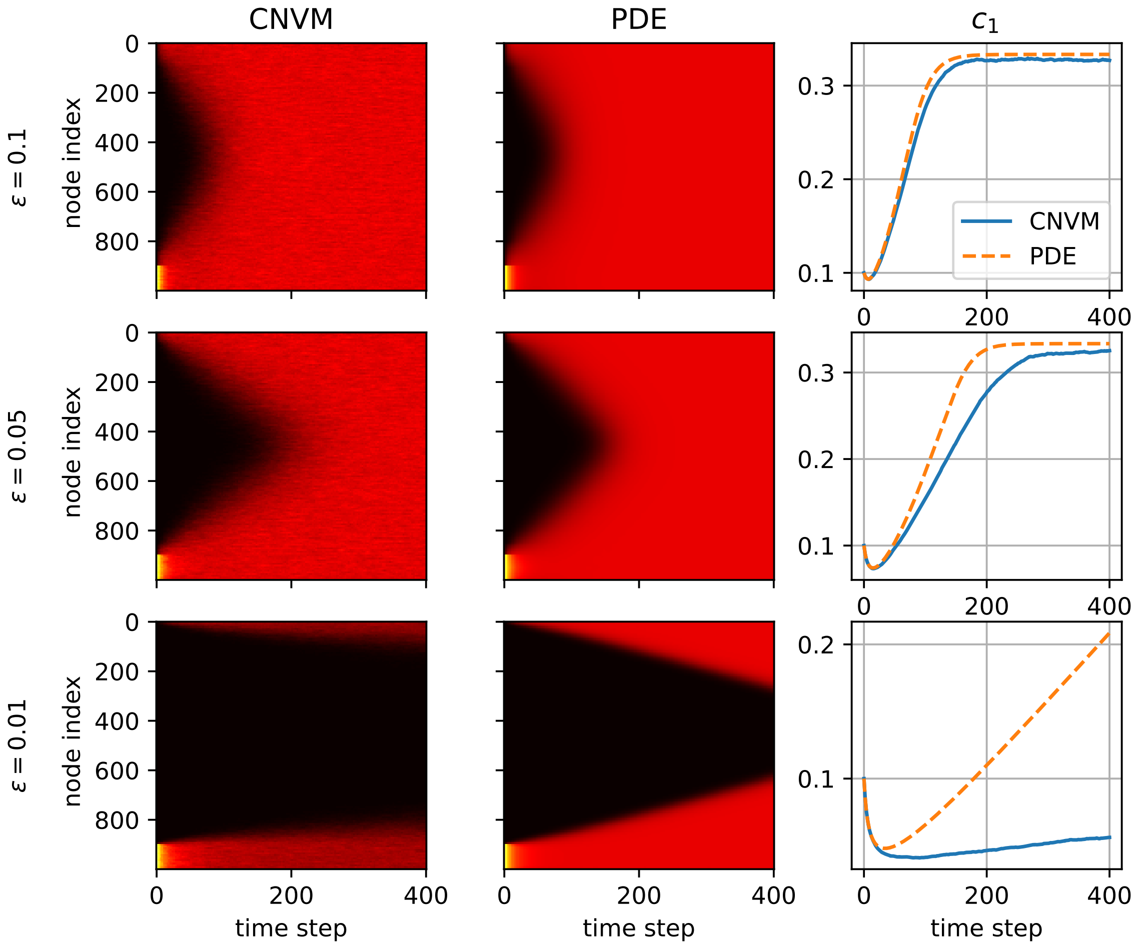


Figure 7.6.: Comparison of SIS dynamics on a ring with $N = 1000$ and $K = 2\varepsilon N$ (left column) with graphon PDE mean-field limit (7.20) (central column), for different ε (rows). The heatmaps depict for each node the probability of being infected, yellow = 1, black = 0. The right column shows the average share c_1 of infected nodes over time.

limit does not match the model at all. The mean-field limit predicts a much larger rate at which the infection spreads on the ring than the actual spreading rate observed in model simulations. This is likely due to the low number of neighbors in the finite-size network hindering the spreading of the infection that is reliant on the occurrence of several discrete infection events, whereas in the "infinite-size" mean-field PDE infection is a smooth process that occurs instantly. Again, recalling the requirements of *indistinguishability* and *interchangeability* for mean-field theory this result is not surprising; although for the graphon mean-field limit they do not have to be fulfilled globally like before, but instead locally around each "node" $x \in [0, 1]$.

8. Conclusion

In this chapter the most relevant contributions of this thesis are recapitulated. Possible extensions and further research questions are discussed. Finally, a more general outlook is given concerning the overall topic of this thesis.

Main contributions. In chapter 4 it was discussed that the standard stochastic simulation algorithm (SSA) for continuous-time Markov chains is not feasible for Markov jump processes on networks due the exponentially scaling number M^N of system states, where N denotes the number of nodes and M the number of node states. The resulting memory requirements make the precomputation of rates and lookup tables, which is necessary for efficient simulation in $\mathcal{O}(N)$ time, impossible. As a consequence, in the standard SSA the needed rates have to be computed during the simulation, leading to an unwieldy time complexity of $\mathcal{O}(N^2 M)$. However, as nothing is precomputed, the memory requirements are mild: only the network data of size $\mathcal{O}(\sum_i^N d_i)$, where d_i is the degree of node i, has to be stored. The modified version of the SSA for the continuous-time noisy voter model (CNVM) that was presented in this chapter (see algorithm 4) achieves both fast simulation in $\mathcal{O}(N)$ time and low memory requirements of $\mathcal{O}(\sum_i^N d_i)$ by exploiting the specific structure of the transition rates in the CNVM. This efficient simulation algorithm made many numerical examples discussed in this thesis and the resulting insights possible. It also plays an important role in the data-driven method for learning collective variables from chapter 6, which requires the execution of simulations for data acquisition.

The convergence of Markov jump processes on networks to a mean-field limit was discussed in chapter 5. The main theorem 5.2 provides conditions under which the stochastic dynamics of the *shares* of each node state concentrates around a deterministic mean-field equation (MFE). It is a generalization of a well-known concentration result for well-mixed systems to the setting of random graphs. Recall that both the setting of sampling graph and dynamics simultaneously (annealed result), and the setting of first sampling a sequence of graphs and then the dynamics (quenched result, see corollary 5.4) were discussed. Moreover, it was shown that the error between model and MFE never decreases faster than $N^{-1/2}$ due to the central limit theorem, but can decrease arbitrarily slowly depending on the dynamical model and network, see proposition 5.7.

In section 5.2 the main theorem was utilized to derive parameter bounds for several random graph models that guarantee convergence to the mean-field limit for the CNVM. More specifically, it was shown that the edge density p in Erdős–Rényi random graphs has to dominate $\log(N)/N$ asymptotically. In the stochastic block model, for each cluster k there has to be a cluster k' such that the edge probability $p_{k,k'}$ dominates $\log(N)/N$, and only the asymptotically largest cluster to cluster connections remain present in the mean-field equation. For the uniformly random d-regular graph, even an arbitrarily slow convergence of the degree d to infinity already yields convergence to the mean-field limit.

Another main contribution of this thesis is the data-driven method for learning collective variables (CVs) discussed in chapter 6. Instead of relying on guesswork for the discovery of CVs, this method is able to explicitly test dynamical conditions for the existence of good CVs during the data-driven computation, i.e., whether the transition density functions exhibit a low-dimensional submanifold structure, and automatically extract the best choice of CVs. The particular design of the ansatz functions in the regression step allows to assess the relationship between the learned CVs and topological features of the network, making the CVs easy to interpret. This was demonstrated in several examples. In the case of a stochastic block model network the learned CVs essentially measure the state shares in each cluster, which is known to be an asymptotically optimal choice of CVs (see section 5.2.3). For a ring network, the method indicated that no definite set of low-dimensional good CVs exist, but instead increasing the dimension allows a progressively better resolution of the state density as a function on the ring, which also agrees well with the literature. For scale-free networks generated by the Albert–Barabási model, a one-dimensional CV was learned that describes the *degree-weighted* state shares in the system, which is a novel CV in the sense that an associated reduced system has not been discussed in the literature. This reduced system utilizing the degree-weighted shares was investigated in section 6.3 using a common data-driven technique for learning differential equations. It became apparent that a good approximation is systematically given by a modification of the standard mean-field equation that employs slightly reduced interaction rates compared to the actual model parameters. However, more thorough testing is required to verify this observation and a theoretical justification is still missing.

Finally, the notion of complementing moment closure methods for discrete-state systems on networks with data-driven methods to minimize the error of the closure constitutes another contribution of this thesis. This idea was successfully demonstrated in section 7.2 for a triplet approximation of the CNVM on a ring network. The learned optimal closure was able to reduce the error of the original triplet approximation considerably, and this technique was significantly more robust than learning a complete dynamical model from scratch.

Contributions inspiring further work. In the following, the smaller contributions that have been presented in this thesis are summarized. They would often require some further work or extensions to make a significant impact. Some possibilities are outlined below.

In section 5.3, the mean-field equation was combined with a stochastic term such that the resulting SDE delivers a viable approximation even for medium-sized populations, which was verified via numerical experiments. However, a precise analysis of the resulting approximation error in dependence on the dynamical model and the underlying network topology was not conducted and could be the subject of future work.

Similarly, the hybrid model in the form of a piecewise-deterministic Markov process, which was derived in section 5.4 as the large population limit of leader-follower dynamics, was only tested numerically and would benefit from a theoretical analysis of its approximation error. Moreover, the network employed in the numerical example was created rather artificially by inserting “leader nodes” with maximum connectivity. Instead, it would be interesting to consider network models that naturally generate such high degree leader nodes, for instance the *Bianconi–Barabási model* [171] in which a modified preferential attachment method creates nodes that attract a large portion of the links.

Furthermore, the results of the data-driven method to learn CVs that have been presented in section 6.2 also inspire further investigation. For instance, it has been shown that in the case of two sparsely connected network communities the shares have to be measured separately for each community. It would be interesting to study how the transition manifold and associated CVs transform from being two-dimensional to one-dimensional as one increases the edge density between the communities. Moreover, recall that for the ring the coordinates of the learned CV represented Fourier modes of the state density function on the ring. After deriving or learning a reduced model in these coordinates, it would be interesting to compare it to the triplet approximation on the ring from section 7.2 in future work, in order to assess which set of CVs is able to produce an approximate model with smaller error.

Lastly, in section 7.1 it was examined numerically how the approximation error of the mean-field equation decreases from large to small when interpolating from a ring network through the small-world regime to an Erdős–Rényi random graph. Deriving an analytical estimate of this error could elucidate how exactly certain network properties impact how well a system can be approximated by mean-field theory and when convergence in the large population limit occurs.

Outlook. Although the main theorem 5.2 was successfully applied to prove convergence to the mean-field limit for several examples, its practicality is limited due to the requirement of having to choose classes and reduced propensity functions beforehand. A desirable long-term objective would be to find a more general theorem that instead states explicit conditions on the dynamical model and the network.

It is also not well-understood yet which properties of dynamical model and network lead to the occurrence of concentration effects in the sense that the evolution of state shares (or some other relevant collective variables) becomes deterministic in the large population limit. This phenomenon has been observed for systems that converge to the mean-field limit, e.g., the CNVM on Erdős–Rényi random graphs, but also for systems that do not, e.g., the CNVM on Albert–Barabási networks. For some systems, like the CNVM on a star graph, a concentration effect can not be observed at all as the state shares remain stochastic even in the large population limit. In the cases where the shares concentrate but the limiting dynamics is not given by the standard mean-field equation, it is not clear if a different ODE exists that constitutes an exact limit and, if so, which collective variables it uses and how to derive it.

The problems and questions mentioned above will certainly not be easy to tackle. An important and useful tool to elevate our understanding of these issues is given by data-driven methods like the technique for learning collective variables presented in this thesis. They will likely play a vital role in advancing the field by enabling us to systematically and automatically form and test new hypotheses, hopefully inspiring novel analytical approaches and theory.

A. Appendix

This appendix contains several auxiliary lemmas that are used throughout the thesis.

Lemma A.1 (Chernoff bound)**.** *Let $\boldsymbol{X}_1, \dots, \boldsymbol{X}_n$ be independent random variables with values in $\{0,1\}$ and denote $\boldsymbol{X} \coloneqq \sum_i \boldsymbol{X}_i$. Then for all $\varepsilon > 0$*

$$\mathbb{P}\big(\,|\boldsymbol{X} - \mathbb{E}[\boldsymbol{X}]| \geq \varepsilon\big) \leq 2\exp\left(-\frac{\varepsilon^2}{3\ \mathbb{E}[\boldsymbol{X}]}\right). \tag{A.1}$$

Proof. See for example [172, Corollary 4.6]. □

Lemma A.2. *Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space and $\boldsymbol{z}^\ell : \Omega \times \mathbb{R} \to [0,1]$, $\ell \in \mathbb{N}$, a sequence of stochastic processes with Lebesgue-measurable realizations, i.e., $\boldsymbol{z}^\ell(\omega, \cdot)$ is measurable for all ω. We denote the random variable describing the process at time $t \in \mathbb{R}$ as $\boldsymbol{z}^\ell(t)$. Assume that for all $\varepsilon > 0$ there exists a function $f_\varepsilon : \mathbb{N} \to \mathbb{R}$ such that*

a) $\forall \ell \in \mathbb{N}\ \forall s \in \mathbb{R} : \mathbb{P}(\boldsymbol{z}^\ell(s) \geq \varepsilon) \leq f_\varepsilon(\ell)$

b) $\lim_{\ell \to \infty} f_\varepsilon(\ell) = 0$.

Then it follows that

1) $\forall s \in \mathbb{R} : \lim_{\ell \to \infty} \mathbb{E}[\boldsymbol{z}^\ell(s)] = 0$

2) $\forall t \in \mathbb{R}_{\geq 0} : \lim_{\ell \to \infty} \mathbb{E}[\int_0^t \boldsymbol{z}^\ell(s) ds] = 0$

3) $\forall t \in \mathbb{R}_{\geq 0} : \int_0^t \boldsymbol{z}^\ell(s) ds \xrightarrow{\text{p}} 0$ *as* $\ell \to \infty$.

Proof. Let $\varepsilon > 0$ and define for every $\ell \in \mathbb{N}$ the stochastic process $\hat{\boldsymbol{z}}^\ell(s)$ by

$$\hat{\boldsymbol{z}}^\ell(s) \coloneqq \begin{cases} \varepsilon, & \boldsymbol{z}^\ell(s) < \varepsilon \\ 1, & \text{else.} \end{cases} \tag{A.2}$$

Thus, $\boldsymbol{z}^\ell(s) \leq \hat{\boldsymbol{z}}^\ell(s)$ and

$$\begin{aligned} \mathbb{E}[\boldsymbol{z}^\ell(s)] \leq \mathbb{E}[\hat{\boldsymbol{z}}^\ell(s)] &= \varepsilon \mathbb{P}(\boldsymbol{z}^\ell(s) < \varepsilon) + \mathbb{P}(\boldsymbol{z}^\ell(s) \geq \varepsilon) && \text{(A.3)} \\ &\leq \varepsilon + f_\varepsilon(\ell). && \text{(A.4)} \end{aligned}$$

By assumption b) this yields $\lim_{\ell \to \infty} \mathbb{E}[\boldsymbol{z}^\ell(s)] \leq \varepsilon$. As $\varepsilon > 0$ was arbitrary, statement 1) follows.

In order to prove statement 2), Tonelli's theorem is applied to interchange integral and expected value, i.e.

$$\mathbb{E}\Big[\int_0^t \boldsymbol{z}^\ell(s) ds\Big] = \int_0^t \mathbb{E}[\boldsymbol{z}^\ell(s)] ds, \tag{A.5}$$

since the integrand is non-negative. As $\mathbb{E}[\boldsymbol{z}^\ell(s)] \leq 1$, the dominated convergence theorem can be applied, which yields

$$\lim_{\ell\to\infty} \int_0^t \mathbb{E}[\boldsymbol{z}^\ell(s)]ds = \int_0^t \lim_{\ell\to\infty} \mathbb{E}[\boldsymbol{z}^\ell(s)]ds = 0 \tag{A.6}$$

by statement 1). Statement 3) follows directly from 2), as convergence in L^1 is stronger than convergence in probability. □

Lemma A.3. *The symbols used in this lemma are defined in section 3.1.6. Let* $1 \leq b \leq Nd$ *and for a tuple* $t \in \Pi$ *let* $h(t) := |\{(s,e) \in \psi(t) \mid s \leq b < e\}|$ *denote the number of edges that cross the boundary* b*. Let* $t, t' \in \Pi$ *only differ in one coordinate* l*, i.e.,* $t = (t_1, \dots, t_\eta)$*,* $t' = (t_1, \dots, t_{l-1}, t'_l, t_{l+1}, \dots, t_\eta)$*. Then it follows that*

$$|h(t) - h(t')| \leq 2. \tag{A.7}$$

Proof. Define

$$\delta^r := \begin{cases} 1, & \text{if } u_0^r \leq b < u_{t_r}^r \\ 0, & \text{else} \end{cases} \tag{A.8}$$

and note that $h(t) = \sum_{r=1}^{\eta} \delta^r$. Moreover, let $i^r := \max\{i \mid u_i^r \leq b\}$ be the index of the largest element in U^r that is not larger than b. (Assign $\max \emptyset := -1$.) The following relations between δ^r and i^r hold:

$$0 \leq i^r < t_r \quad \Leftrightarrow \quad \delta^r = 1 \tag{A.9}$$

$$i^r \geq t_r \text{ or } i^r = -1 \quad \Leftrightarrow \quad \delta^r = 0 \tag{A.10}$$

$$\delta^r = 1 \quad \Rightarrow \quad i^{r+1} = i^r - 1 \tag{A.11}$$

$$\delta^r = 0 \quad \Rightarrow \quad i^{r+1} = \begin{cases} -1, & \text{if } i^r = -1 \\ i^r - 2, & \text{else.} \end{cases} \tag{A.12}$$

Relation (A.11) holds because $\delta^r = 1$ implies that exactly one element that is smaller or equal to b is removed from U^r, and hence i^{r+1} is one less than i^r. Relation (A.12) holds because $\delta^r = 0$ implies that either two elements that are smaller or equal to b are removed from U^r, which yields a reduction of i^{r+1} by 2 compared to i^r, or both removed elements are larger than b, which is the case if $i^r = -1$.

Let the respective analogous objects for t' be denoted as $(U')^r$, $(u')_i^r$, $(\delta')^r$, and $(i')^r$. W.l.o.g., it is assumed that $t_l < t'_l$.

Clearly, for $r \leq l$ it follows that $U^r = (U')^r$ and $i^r = (i')^r$, and for $r < l$ it follows that $\delta^r = (\delta')^r$. Consider the case that it also holds $\delta^l = (\delta')^l$. Due to equations (A.11) and (A.12), this yields $i^{l+1} = (i')^{l+1}$, and as $t_r = t'_r$ for $r > l$, it is $\delta^{l+1} = (\delta')^{l+1}$ via equations (A.9) and (A.10). By iteration, this implies $\delta^r = (\delta')^r$ for all $r > \ell$ and hence $h(t) = h(t')$.

Now consider the case that $\delta^l \neq (\delta')^l$, i.e., by $t_l < t'_l$ and (A.8), $\delta^l = 0$ and $(\delta')^l = 1$. Due to (A.10) this implies $i^l = (i')^l \neq -1$. Also, by equations (A.11) and (A.12) it follows that $(i')^{l+1} = i^{l+1} + 1$. Depending on the value of $(i')^{l+1}$, one of the following three outcomes occurs:

1. $(i')^{l+1} = 0$: It follows from (A.9) that $(\delta')^{l+1} = 1$, and because $i^{l+1} = -1$ it is $\delta^{l+1} = 0$. Moreover, in the next step from (A.11) it follows that $(i')^{l+2} = -1 = i^{l+2}$ and thus, $(i')^r = i^r$ also for all subsequent steps $r > l + 2$. As a result, for all $r > (l + 1)$ it is $\delta^r = (\delta')^r = 0$. All in all, this yields $h(t') = h(t) + 2$.

2. $(i')^{l+1} = t_{l+1}$: It follows from (A.10) that $(\delta')^{l+1} = 0$, and because $i^{l+1} = t_{l+1} - 1$ it is $\delta^{l+1} = 1$. Hence, by equations (A.11) and (A.12) it follows that $(i')^{l+2} = i^{l+2}$ and thus, by iteration also $(i')^r = i^r$ for all subsequent steps $r > l+2$. As a result, for all $r > (l+1)$ it is $\delta^r = (\delta')^r$. All in all, this yields $h(t') = h(t) + 1 - 1 = h(t)$.

3. Else: For all other values of $(i')^{l+1}$, it is $\delta^{l+1} = (\delta')^{l+1}$. Thus, in the next step the relation $(i')^{l+2} = i^{l+2} + 1$ is still true due to (A.11) if $\delta^{l+1} = 1$, or due to (A.12) if $\delta^{l+1} = 0$. By iteration, either case 1. or 2. occur in one of the subsequent steps $r > (l+1)$, i.e., either $(i')^r = 0$ or $(i')^r = t_r$, which yields $h(t') = h(t) + 2$, or $h(t') = h(t)$ respectively.

□

Zusammenfassung

Dynamische Systeme auf Netzwerken werden häufig verwendet, um Systeme zu modellieren, die aus vielen interagierenden Einheiten, welche *Agenten* genannt werden, bestehen. Hierbei stellen die Knoten im Netzwerk die Agenten dar, die Kanten repräsentieren die Beziehungen zwischen den Agenten, und der *Zustand* jedes Agenten entwickelt sich im Laufe der Zeit in Abhängigkeit von den Zuständen seiner Nachbarn, in der Regel auf stochastische Weise. Obwohl die Zustandsentwicklung jedes einzelnen Agenten oft durch einfache Regeln und Mechanismen bestimmt wird, kann das *kollektive* oder *emergente* Verhalten des gesamten Systems, welches das Ergebnis vieler individueller Interaktionen ist, sehr schwer zu antizipieren und zu verstehen sein. Die Untersuchung dieses kollektiven Verhaltens ist der Schwerpunkt dieser Arbeit.

Obwohl das kollektive Verhalten schwer vorherzusagen ist, ist es in der Regel deutlich weniger komplex als es die große Anzahl von Freiheitsgraden zulassen würde und folgt stattdessen (ungefähr) einer niedrigdimensionalen Dynamik. Das Verständnis des kollektiven Verhaltens besteht daher aus zwei Schritten. Erstens muss eine Projektion gefunden werden, die den hochdimensionalen *mikroskopischen* Systemzustand, der den Zustand jedes einzelnen Agenten enthält, auf einen niedrigdimensionalen *makroskopischen* Systemzustand abbildet, der nur die wesentliche aggregierte Information zur Beschreibung des kollektiven Verhaltens enthält. Diese Projektion, die unnötige Freiheitsgrade und schnell abklingende Prozesse des ursprünglichen Systems herausfiltert, wird als *kollektive Variable* bezeichnet. Zweitens muss das reduzierte makroskopische System hergeleitet werden. Wenn die Wahl der kollektiven Variable angemessen war, ist das makroskopische System in der Lage die niedrigdimensionale Projektion des ursprünglichen Modells, also das kollektive Verhalten, zu reproduzieren. Ähnlich wie beim Gesetz der großen Zahlen führen die aggregierten zufälligen Aktionen der vielen Agenten manchmal zu einer annähernd deterministischen makroskopischen Dynamik, was als *Konzentrationseffekt* bezeichnet wird.

Diese Arbeit befasst sich hauptsächlich mit gedächtnislosen dynamischen Systemen mit diskretem Zustandsraum auf (zufälligen) Netzwerken und behandelt die effiziente Simulation, die Herleitung von kollektiven Variablen und makroskopischer Dynamik sowie das Auftreten von Konzentrationseffekten. Für solche Systeme stellen die *Anteile* jedes diskreten Zustands im System (oder in bestimmten Teilsystemen) eine wichtige Wahl der kollektiven Variablen dar. Es werden Bedingungen nachgewiesen, welche die Konvergenz der Dynamik dieser Anteile zu einer deterministischen „mean-field“ Differentialgleichung im Limit unendlich vieler Agenten garantieren. Diese Bedingungen ermöglichen die Herleitung von Parameterschranken für populäre Zufallsgraphenmodelle, zum Beispiel Erdős–Rényi Zufallsgraphen, das stochastische Blockmodell und reguläre Zufallsgraphen, sodass die Konvergenz zum „mean-field“ Limit gewährleistet ist. Dies wird für ein zeitkontinuierliches „voter model“ demonstriert. Für Systeme, die keine Konvergenz zum „mean-field“ Limit aufweisen und für welche die oben genannten Anteile keine geeigneten kollektiven Variablen sind, wird eine datengesteuerte Methode zum algorithmischen Lernen interpretierbarer kollektiver Variablen aus Modellsimulationen vorgestellt. Diese Methode ermöglicht es die Qualität der gelernten kollektiven Variablen zu bewerten und ihre Beziehung zu topologischen Merkmalen des Netzwerks abzuleiten. In Kombination mit etablierten Techniken zum Lernen von Dynamik aus Daten kann eine automatische Evaluation des kollektiven Verhaltens erreicht werden. Dies wird beispielhaft für das „voter model“ auf skalenfreien Netzwerken demonstriert.

Bibliography

[1] M. Lücke, J. Heitzig, P. Koltai, N. Molkenthin, and S. Winkelmann. Large population limits of Markov processes on random networks. *Stochastic Processes and their Applications*, 166:104220, 2023.

[2] M. Lücke, S. Winkelmann, J. Heitzig, N. Molkenthin, and P. Koltai. Learning interpretable collective variables for spreading processes on networks. *Physical Review E*, 109(2):l022301, 2024.

[3] M. Lücke, P. Koltai, S. Winkelmann, N. Molkenthin, and J. Heitzig. Discovering collective variable dynamics of agent-based models. In *Extended Abstracts presented at the 25th International Symposium on Mathematical Theory of Networks and Systems MTNS 2022*, pages 202–205. University of Bayreuth, 2022.

[4] E. Bonabeau. Agent-based modeling: methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences*, 99:7280–7287, 2002.

[5] D. Heard, G. Dent, T. Schifeling, and D. Banks. Agent-based models and microsimulation. *Annual Review of Statistics and Its Application*, 2(1):259–272, 2015.

[6] C. Macal and M. North. Introductory tutorial: agent-based modeling and simulation. In *Proceedings of the Winter Simulation Conference 2014*. IEEE, 2014.

[7] S. J. E. Taylor, editor. Agent-Based Modeling and Simulation. Palgrave Macmillan UK, 2014.

[8] M. Porter and J. Gleeson. Dynamical Systems on Networks. Springer International Publishing, 2016.

[9] A. Barrat, M. Barthélemy, and A. Vespignani. Dynamical Processes on Complex Networks. Cambridge University Press, 2008.

[10] T. M. Liggett. Interacting Particle Systems. Springer Berlin Heidelberg, 2005.

[11] S. Majhi, M. Perc, and D. Ghosh. Dynamics on higher-order networks: a review. *Journal of The Royal Society Interface*, 19(188), 2022.

[12] T. Gross and B. Blasius. Adaptive coevolutionary networks: a review. *Journal of The Royal Society Interface*, 5(20):259–271, 2007.

[13] P. Holme and J. Saramäki. Temporal networks. *Physics Reports*, 519(3):97–125, 2012.

[14] L. Horstmeyer and C. Kuehn. Adaptive voter model on simplicial complexes. *Physical Review E*, 101(2):022305, 2020.

[15] B. Barzel, Y.-Y. Liu, and A.-L. Barabási. Constructing minimal models for complex system dynamics. *Nature Communications*, 6(1), 2015.

[16] B. Barzel and A.-L. Barabási. Universality in network dynamics. *Nature Physics*, 9(10):673–681, 2013.

[17] Y. Kuramoto. Self-entrainment of a population of coupled non-linear oscillators. In *International Symposium on Mathematical Problems in Theoretical Physics*, pages 420–422. Springer-Verlag, 1975.

[18] F. A. Rodrigues, T. K. D. Peron, P. Ji, and J. Kurths. The Kuramoto model in complex networks. *Physics Reports*, 610:1–98, 2016.

[19] M. H. DeGroot. Reaching a consensus. *Journal of the American Statistical Association*, 69(345):118–121, 1974.

[20] B. Golub and M. O. Jackson. Naïve learning in social networks and the wisdom of crowds. *American Economic Journal: Microeconomics*, 2(1):112–149, 2010.

[21] M. McPherson, L. Smith-Lovin, and J. M. Cook. Birds of a feather: homophily in social networks. *Annual Review of Sociology*, 27(1):415–444, 2001.

[22] G. Deffuant, D. Neau, F. Amblard, and G. Weisbuch. Mixing beliefs among interacting agents. *Advances in Complex Systems*, 03:87–98, 2000.

[23] C. Castellano, S. Fortunato, and V. Loreto. Statistical physics of social dynamics. *Reviews of Modern Physics*, 81(2):591–646, 2009.

[24] R. Hegselmann and U. Krause. Opinion dynamics and bounded confidence: models, analysis and simulation. *Journal of Artificial Societies and Social Simulation*, 5(3), 2002.

[25] M. A. S. Kolarijani, A. V. Proskurnikov, and P. M. Esfahani. Macroscopic noisy bounded confidence models with distributed radical opinions. *IEEE Transactions on Automatic Control*, 66(3):1174–1189, 2021.

[26] P. Clifford and A. Sudbury. A model for spatial conflict. *Biometrika*, 60(3):581–588, 1973.

[27] R. A. Holley and T. M. Liggett. Ergodic theorems for weakly interacting infinite systems and the voter model. *The Annals of Probability*, 3(4), 1975.

[28] J. Fernández-Gracia, K. Suchecki, J. J. Ramasco, M. S. Miguel, and V. M. Eguíluz. Is the voter model a model for voters? *Physical Review Letters*, 112(15):158701, 2014.

[29] M. Granovetter. Threshold models of collective behavior. *American Journal of Sociology*, 83(6):1420–1443, 1978.

[30] D. J. Watts. A simple model of global cascades on random networks. *Proceedings of the National Academy of Sciences*, 99(9):5766–5771, 2002.

[31] S. Galam. Minority opinion spreading in random geometry. *The European Physical Journal B*, 25(4):403–406, 2002.

[32] P. L. Krapivsky and S. Redner. Dynamics of majority rule in two-state interacting spin systems. *Physical Review Letters*, 90(23):238701, 2003.

[33] R. Burioni and D. Cassi. Random walks on graphs: ideas, techniques and results. *Journal of Physics A: Mathematical and General*, 38(8):R45–R78, 2005.

[34] J. D. Noh and H. Rieger. Random walks on complex networks. *Physical Review Letters*, 92(11):118701, 2004.

[35] A. Das Sarma, A. R. Molla, G. Pandurangan, and E. Upfal. Fast distributed pagerank computation. In *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2013, pages 11–26.

[36] I. Neri, N. Kern, and A. Parmeggiani. Totally asymmetric simple exclusion process on networks. *Physical Review Letters*, 107(6):068702, 2011.

[37] S. Carmi, Z. Wu, S. Havlin, and H. E. Stanley. Transport in networks with multiple sources and sinks. *EPL (Europhysics Letters)*, 84(2):28005, 2008.

[38] B. Tadić, G. J. Rodgers, and S. Thurner. Transport on complex networks: flow, jamming and optimization. *International Journal of Bifurcation and Chaos*, 17(07):2363–2385, 2007.

[39] I. Z. Kiss, J. C. Miller, and P. L. Simon. Mathematics of Epidemics on Networks. Springer International Publishing, 2017.

[40] S. Winkelmann and C. Schütte. Stochastic Dynamics in Computational Biology. Springer International Publishing, 2020.

[41] D. T. Gillespie. Approximate accelerated stochastic simulation of chemically reacting systems. *The Journal of Chemical Physics*, 115(4):1716–1733, 2001.

[42] W. J. Anderson. Continuous-Time Markov Chains. Springer New York, 1991.

[43] A. Carro, R. Toral, and M. S. Miguel. The noisy voter model on complex networks. *Scientific Reports*, 6(1), 2016.

[44] S. Hayward and N. Go. Collective variable description of native protein dynamics. *Annual Review of Physical Chemistry*, 46(1):223–250, 1995.

[45] F. Sittel and G. Stock. Perspective: identification of collective variables and metastable states of protein dynamics. *The Journal of Chemical Physics*, 149(15), 2018.

[46] C. Wehmeyer and F. Noé. Time-lagged autoencoders: deep learning of slow collective variables for molecular kinetics. *The Journal of Chemical Physics*, 148(24), 2018.

[47] A. Bovier and F. den Hollander. Metastability. Springer International Publishing, 2015.

[48] P. Metzner, C. Schütte, and E. Vanden-Eijnden. Transition path theory for Markov jump processes. *Multiscale Modeling & Simulation*, 7(3):1192–1219, 2009.

[49] N. D. Socci, J. N. Onuchic, and P. G. Wolynes. Diffusive dynamics of the reaction coordinate for protein folding funnels. *The Journal of Chemical Physics*, 104(15):5860–5868, 1996.

[50] R. B. Best and G. Hummer. Reaction coordinates and rates from transition paths. *Proceedings of the National Academy of Sciences*, 102(19):6732–6737, 2005.

[51] P. Maity, A. Bittracher, P. Koltai, and J. Schumacher. Collective variables between large-scale states in turbulent convection. *Physical Review Research*, 5(3):033061, 2023.

[52] L. D. Smith and G. A. Gottwald. Model reduction for the collective dynamics of globally coupled oscillators: from finite networks to the thermodynamic limit. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 30(9), 2020.

[53] A. Pikovsky and M. Rosenblum. Dynamics of heterogeneous oscillator ensembles in terms of collective variables. *Physica D: Nonlinear Phenomena*, 240(9-10):872–881, 2011.

[54] L. Helfmann, J. Heitzig, P. Koltai, J. Kurths, and C. Schütte. Statistical analysis of tipping pathways in agent-based models. *The European Physical Journal Special Topics*, 230(16-17):3249–3271, 2021.

[55] J.-H. Niemann. Learning Reduced Models for Large-Scale Agent-Based Systems. PhD thesis, 2022.

[56] A. Bittracher, P. Koltai, S. Klus, R. Banisch, M. Dellnitz, and C. Schütte. Transition manifolds of complex metastable systems: theory and data-driven computation of effective dynamics. *Journal of Nonlinear Science*, 28(2):471–512, 2017.

[57] A. Bittracher, S. Klus, B. Hamzi, P. Koltai, and C. Schütte. Dimensionality reduction of complex metastable systems via kernel embeddings of transition manifolds. *Journal of Nonlinear Science*, 31(1), 2020.

[58] A. Bittracher, M. Mollenhauer, P. Koltai, and C. Schütte. Optimal reaction coordinates: variational characterization and sparse computation. *Multiscale Modeling & Simulation*, 21(2):449–488, 2023.

[59] R. Zwanzig. Nonequilibrium Statistical Mechanics. Oxford University PressNew York, NY, 2001.

[60] S. Klus, F. Nüske, P. Koltai, H. Wu, I. Kevrekidis, C. Schütte, and F. Noé. Data-driven model reduction and transfer operator approximation. *Journal of Nonlinear Science*, 28(3):985–1010, 2018.

[61] P. Deuflhard, W. Huisinga, A. Fischer, and C. Schütte. Identification of almost invariant aggregates in reversible nearly uncoupled Markov chains. *Linear Algebra and its Applications*, 315(1–3):39–59, 2000.

[62] C. Schütte and M. Sarich. Metastability and Markov State Models in Molecular Dynamics. American Mathematical Society, 2013.

[63] P. Deuflhard and M. Weber. Robust Perron cluster analysis in conformation dynamics. *Linear Algebra and its Applications*, 398:161–184, 2005.

[64] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley. A data–driven approximation of the Koopman operator: extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25(6):1307–1346, 2015.

[65] G. Froyland, G. A. Gottwald, and A. Hammerlindl. A computational method to extract macroscopic variables and their dynamics in multiscale systems. *SIAM Journal on Applied Dynamical Systems*, 13(4):1816–1846, 2014.

[66] G. Pérez-Hernández, F. Paul, T. Giorgino, G. De Fabritiis, and F. Noé. Identification of slow molecular order parameters for Markov model construction. *The Journal of Chemical Physics*, 139(1), 2013.

[67] A. Mardt, L. Pasquali, H. Wu, and F. Noé. Vampnets for deep learning of molecular kinetics. *Nature Communications*, 9(1), 2018.

[68] F. Noé and F. Nüske. A variational approach to modeling slow processes in stochastic dynamical systems. *Multiscale Modeling & Simulation*, 11(2):635–655, 2013.

[69] R. J. Rabben, S. Ray, and M. Weber. Isokann: invariant subspaces of Koopman operators learned by a neural network. *The Journal of Chemical Physics*, 153(11), 2020.

[70] K. Champion, B. Lusch, J. N. Kutz, and S. L. Brunton. Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences*, 116(45):22445–22451, 2019.

[71] W. Yue, L. D. Smith, and G. A. Gottwald. Model reduction for the Kuramoto-Sakaguchi model: the importance of nonentrained rogue oscillators. *Physical Review E*, 101(6):062213, 2020.

[72] E. Ising. Beitrag zur Theorie des Ferromagnetismus. *Zeitschrift für Physik*, 31(1):253–258, 1925.

[73] B. D. Goddard, B. Gooding, H. Short, and G. A. Pavliotis. Noisy bounded confidence models for opinion dynamics: the effect of boundary conditions on phase transitions. *IMA Journal of Applied Mathematics*, 87(1):80–110, 2021.

[74] M. Taylor, P. L. Simon, D. M. Green, T. House, and I. Z. Kiss. From Markovian to pairwise epidemic models and the performance of moment closure approximations. *Journal of Mathematical Biology*, 64(6):1021–1042, 2011.

[75] A. F. Peralta, A. Carro, M. S. Miguel, and R. Toral. Stochastic pair approximation treatment of the noisy voter model. *New Journal of Physics*, 20(10):103045, 2018.

[76] P. Moretti, S. Y. Liu, A. Baronchelli, and R. Pastor-Satorras. Heterogenous mean-field analysis of a generalized voter-like model on networks. *The European Physical Journal B*, 85(3), 2012.

[77] E. Pugliese and C. Castellano. Heterogeneous pair approximation for voter models on networks. *EPL (Europhysics Letters)*, 88(5):58004, 2009.

[78] M. J. Keeling, D. A. Rand, and A. J. Morris. Correlation models for childhood epidemics. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 264(1385):1149–1156, 1997.

[79] N. Ayi and N. Pouradier Duteil. Mean-field and graph limits for collective dynamics models with time-varying weights. *Journal of Differential Equations*, 299:65–110, 2021.

[80] J. P. Gleeson, S. Melnik, J. A. Ward, M. A. Porter, and P. J. Mucha. Accuracy of mean-field theory for dynamics on real-world networks. *Physical Review E*, 85(2):026106, 2012.

[81] L. Zhao and X. Xue. The voter model with a slow membrane. *Journal of Theoretical Probability*, 2024.

[82] W.-T. L. Fan. Stochastic PDEs on graphs as scaling limits of discrete interacting systems. *Bernoulli*, 27(3), 2021.

[83] R. Durrett and W.-T. L. Fan. Genealogies in expanding populations. *The Annals of Applied Probability*, 26(6), 2016.

[84] E. Presutti and H. Spohn. Hydrodynamics of the voter model. *The Annals of Probability*, 11(4), 1983.

[85] D. Keliger, I. Horváth, and B. Takács. Local-density dependent Markov processes on graphons with epidemiological applications. *Stochastic Processes and their Applications*, 148:324–352, 2022.

[86] N. P. Duteil. Mean-field limit of collective dynamics with time-varying weights. *Networks and Heterogeneous Media*, 17(2):129, 2022.

[87] L.-P. Chaintron and A. Diez. Propagation of chaos: a review of models, methods and applications. i. models and methods. *Kinetic and Related Models*, 15(6):895, 2022.

[88] J. Gómez-Serrano, C. Graham, and J.-Y. Le Boudec. The bounded confidence model of opinion dynamics. *Mathematical Models and Methods in Applied Sciences*, 22(02):1150007, 2012.

[89] M. te Vrugt, J. Bickmann, and R. Wittkowski. Effects of social distancing and isolation on epidemic spreading modeled via dynamical density functional theory. *Nature Communications*, 11(1), 2020.

[90] C. Kuehn. Moment closure—a brief review. In *Control of Self-Organizing Nonlinear Systems*. Springer International Publishing, 2016, pages 253–271.

[91] A. R. Vieira, A. F. Peralta, R. Toral, M. S. Miguel, and C. Anteneodo. Pair approximation for the noisy threshold q-voter model. *Physical Review E*, 101(5):052131, 2020.

[92] J. P. Gleeson. High-accuracy approximation of binary-state dynamics on networks. *Physical Review Letters*, 107(6):068701, 2011.

[93] J. P. Gleeson. Binary-state dynamics on complex networks: pair approximation and beyond. *Physical Review X*, 3(2):021004, 2013.

[94] J. G. Kemeny and J. L. Snell. Finite Markov chains. Springer-Verlag, 1976.

[95] P. L. Simon, M. Taylor, and I. Z. Kiss. Exact epidemic models on graphs using graph-automorphism driven lumping. *Journal of Mathematical Biology*, 62(4):479–508, 2010.

[96] W. R. KhudaBukhsh, A. Auddy, Y. Disser, and H. Koeppl. Approximate lumpability for Markovian agent-based models using local symmetries. *Journal of Applied Probability*, 56(3):647–671, 2019.

[97] G. Rubino and G. Rubino. On weak lumpability in Markov chains. *Journal of Applied Probability*, 26(3):446–457, 1989.

[98] T. Dayar and W. J. Stewart. Quasi lumpability, lower-bounding coupling matrices, and nearly completely decomposable Markov chains. *SIAM Journal on Matrix Analysis and Applications*, 18(2):482–498, 1997.

[99] H. De Sterck, T. A. Manteuffel, S. F. McCormick, Q. Nguyen, and J. Ruge. Multilevel adaptive aggregation for Markov chains, with application to web ranking. *SIAM Journal on Scientific Computing*, 30(5):2235–2262, 2008.

[100] D. Knoke and S. Yang. Social Network Analysis. SAGE Publications, Inc., 2020.

[101] C. C. Aggarwal, editor. Social Network Data Analytics. Springer US, 2011.

[102] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee. Measurement and analysis of online social networks. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*. ACM, 2007.

[103] M. Newman. Networks. Oxford University Press, 2018.

[104] A. D. Broido and A. Clauset. Scale-free networks are rare. *Nature Communications*, 10(1), 2019.

[105] M. E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006.

[106] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684), 1998.

[107] M. E. J. Newman, D. J. Watts, and S. H. Strogatz. Random graph models of social networks. *Proceedings of the National Academy of Sciences*, 99:2566–2572, 2002.

[108] P. Erdös and A. Rényi. On the evolution of random graphs. In *The Structure and Dynamics of Networks*, pages 38–82. Princeton University Press, 2011.

[109] A. Frieze and M. Karoński. Introduction to Random Graphs. Cambridge University Press, 2015.

[110] P. W. Holland, K. B. Laskey, and S. Leinhardt. Stochastic blockmodels: first steps. *Social Networks*, 5(2):109–137, 1983.

[111] F. Chung and L. Lu. Connected components in random graphs with given expected degree sequences. *Annals of Combinatorics*, 6(2):125–145, 2002.

[112] F. Chung and L. Lu. The average distances in random graphs with given expected degrees. *Proceedings of the National Academy of Sciences*, 99(25):15879–15882, 2002.

[113] R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47–97, 2002.

[114] B. Bollobás and O. Riordan. The diameter of a scale-free random graph. *Combinatorica*, 24(1):5–34, 2004.

[115] B. Bollobás and O. M. Riordan. Mathematical results on scale-free random graphs. In *Handbook of Graphs and Networks*, pages 1–34. Wiley-VCH Verlag GmbH & Co. KGaA, 2004.

[116] B. Bollobás. A probabilistic proof of an asymptotic formula for the number of labelled regular graphs. *European Journal of Combinatorics*, 1(4):311–316, 1980.

[117] L. Federico and R. Van der Hofstad. Critical window for connectivity in the configuration model. *Combinatorics, Probability and Computing*, 26(5):660–680, 2017.

[118] D. Fernholz and V. Ramachandran. The diameter of sparse random graphs. *Random Structures & Algorithms*, 31(4):482–516, 2007.

[119] B. Bollobás and W. Fernandez de la Vega. The diameter of random regular graphs. *Combinatorica*, 2(2):125–134, 1982.

[120] J. Kim and V. Vu. Sandwiching random graphs: universality between random graph models. *Advances in Mathematics*, 188(2):444–469, 2004.

[121] P. Gao, M. Isaev, and B. McKay. Kim–Vu's sandwich conjecture is true for $d \geq \log^4 n$, 2020. arXiv: `2011.09449 [math.CO]`.

[122] L. Lovász. Large networks and graph limits. American Mathematical Society, 2012.

[123] J.-F. Delmas, D. Dronnier, and P.-A. Zitt. An infinite-dimensional metapopulation SIS model. *Journal of Differential Equations*, 313:1–53, 2022.

[124] E. Bayraktar, S. Chakraborty, and R. Wu. Graphon mean field systems. *The Annals of Applied Probability*, 33(5), 2023.

[125] S. Gao and P. E. Caines. Graphon control of large-scale networks of linear systems. *IEEE Transactions on Automatic Control*, 65(10):4090–4105, 2020.

[126] P. E. Caines and M. Huang. Graphon mean field games and the GMFG equations. In *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018.

[127] I. Benjamini and O. Schramm. Recurrence of distributional limits of finite planar graphs. *Electronic Journal of Probability*, 6, 2001.

[128] C. Borgs, J. Chayes, H. Cohn, and Y. Zhao. An Lp theory of sparse graph convergence I: limits, sparse random graph models, and power law distributions. *Transactions of the American Mathematical Society*, 372(5):3019–3062, 2019.

[129] H. Hatami, L. Lovász, and B. Szegedy. Limits of locally–globally convergent graph sequences. *Geometric and Functional Analysis*, 24(1):269–296, 2014.

[130] Á. Backhausz and B. Szegedy. Action convergence of operators and graphs. *Canadian Journal of Mathematics*, 74(1):72–121, 2020.

[131] C. Kuehn. Network dynamics on graphops. *New Journal of Physics*, 22(5):053030, 2020.

[132] M. A. Gkogkas and C. Kuehn. Graphop mean-field limits for Kuramoto-type models. *SIAM Journal on Applied Dynamical Systems*, 21(1):248–283, 2022.

[133] C. P. Robert and G. Casella. Monte Carlo Statistical Methods. Springer New York, 2004.

[134] D. T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361, 1977.

[135] H.-C. Chen and Y. Asau. On generating random variates from an empirical distribution. *A I I E Transactions*, 6(2):163–166, 1974.

[136] G. Marsaglia, W. W. Tsang, and J. Wang. Fast generation of discrete random variables. *Journal of Statistical Software*, 11(3), 2004.

[137] A. Walker. New fast method for generating discrete random numbers with arbitrary frequency distributions. *Electronics Letters*, 10(8):127, 1974.

[138] D. Keliger. Markov processes on quasi-random graphs. *Acta Mathematica Hungarica*, 2024.

[139] N. Khalil, M. San Miguel, and R. Toral. Zealots in the mean-field noisy voter model. *Physical Review E*, 97(1):012310, 2018.

[140] J.-H. Niemann, S. Klus, and C. Schütte. Data-driven model reduction of agent-based systems using the Koopman generator. *PLOS ONE*, 16(5):e0250970, 2021. R. Grima, editor.

[141] R. Huo and R. Durrett. The zealot voter model. *The Annals of Applied Probability*, 29(5), 2019.

[142] D. D. Chinellato, I. R. Epstein, D. Braha, Y. Bar-Yam, and M. A. M. de Aguiar. Dynamical response of networks under external perturbations: exact results. *Journal of Statistical Physics*, 159(2):221–230, 2015.

[143] T. G. Kurtz. Strong approximation theorems for density dependent Markov chains. *Stochastic Processes and their Applications*, 6(3):223–240, 1978.

[144] D. F. Anderson and T. G. Kurtz. Continuous time Markov chain models for chemical reaction networks. In *Design and Analysis of Biomolecular Circuits*. Springer New York, 2011, pages 3–42.

[145] C. McDiarmid. On the method of bounded differences. In *Surveys in Combinatorics, 1989*. Cambridge University Press, 1989, pages 148–188.

[146] D. T. Gillespie. The chemical Langevin equation. *The Journal of Chemical Physics*, 113(1):297–306, 2000.

[147] L. Helfmann, N. Djurdjevac Conrad, P. Lorenz-Spreen, and C. Schütte. Modelling opinion dynamics under the impact of influencer and media strategies. *Scientific Reports*, 13(1), 2023.

[148] R. Hegselmann and U. Krause. Opinion dynamics under the influence of radical groups, charismatic leaders, and other constant signals: a simple unifying model. *Networks and Heterogeneous Media*, 10(3):477–509, 2015.

[149] S. Pei, J. Wang, F. Morone, and H. A. Makse. Influencer identification in dynamical complex systems. *Journal of Complex Networks*, 8(2), 2019. J. Gomez-Gardenes, editor.

[150] C. A. Moreira, D. M. Schneider, and M. A. M. de Aguiar. Binary dynamics on star networks under external perturbations. *Physical Review E*, 92(4):042812, 2015.

[151] S. Winkelmann and C. Schütte. Hybrid models for chemical reaction networks: multiscale theory and application to gene regulatory systems. *The Journal of Chemical Physics*, 147(11), 2017.

[152] M. H. A. Davis. Piecewise-deterministic Markov processes: a general class of non-diffusion stochastic models. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 46(3):353–376, 1984.

[153] K. Muandet, K. Fukumizu, B. Sriperumbudur, and B. Schölkopf. Kernel mean embedding of distributions: a review and beyond. *Foundations and Trends® in Machine Learning*, 10(1–2):1–141, 2017.

[154] R. R. Coifman and S. Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30, 2006.

[155] R. Coifman, Y. Shkolnisky, F. Sigworth, and A. Singer. Graph laplacian tomography from unknown random projections. *IEEE Transactions on Image Processing*, 17(10):1891–1899, 2008.

[156] J. Gao, B. Barzel, and A.-L. Barabási. Universal resilience patterns in complex networks. *Nature*, 530(7590):307–312, 2016.

[157] E. Laurence, N. Doyon, L. J. Dubé, and P. Desrosiers. Spectral dimension reduction of complex dynamical networks. *Physical Review X*, 9(1):011042, 2019.

[158] R. J. Tibshirani and J. Taylor. The solution path of the generalized lasso. *The Annals of Statistics*, 39(3), 2011.

[159] T. Hastie, R. Tibshirani, and J. Friedman. The Elements of Statistical Learning. Springer New York, 2009.

[160] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd. OSQP: an operator splitting solver for quadratic programs. *Mathematical Programming Computation*, 12(4):637–672, 2020.

[161] J. Reichardt and M. Leone. (un)detectable cluster structure in sparse networks. *Physical Review Letters*, 101(7):078701, 2008.

[162] A. Decelle, F. Krzakala, C. Moore, and L. Zdeborová. Inference and phase transitions in the detection of modules in sparse networks. *Physical Review Letters*, 107(6):065701, 2011.

[163] K. Suchecki, V. M. Eguíluz, and M. S. Miguel. Conservation laws for the voter model in complex networks. *Europhysics Letters (EPL)*, 69(2):228–234, 2005.

[164] G. Bianconi. Mean field solution of the Ising model on a Barabási–Albert network. *Physics Letters A*, 303(2–3):166–168, 2002.

[165] R. Pastor-Satorras and A. Vespignani. Epidemic spreading in scale-free networks. *Physical Review Letters*, 86(14):3200–3203, 2001.

[166] S. L. Brunton, J. L. Proctor, and J. N. Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016.

[167] L. Boninsegna, F. Nüske, and C. Clementi. Sparse learning of stochastic dynamical equations. *The Journal of Chemical Physics*, 148(24), 2018.

[168] D. Lacker, K. Ramanan, and R. Wu. Local weak convergence for sparse networks of interacting processes. *The Annals of Applied Probability*, 33(2), 2023.

[169] T. Franco, P. Gonçalves, and G. M. Schütz. Scaling limits for the exclusion process with a slow site. *Stochastic Processes and their Applications*, 126(3):800–831, 2016.

[170] C. Kuehn. Multiple Time Scale Dynamics. Springer International Publishing, 2015.

[171] G. Bianconi and A.-L. Barabási. Competition and multiscaling in evolving networks. *Europhysics Letters (EPL)*, 54(4):436–442, 2001.

[172] M. Mitzenmacher and E. Upfal. Probability and Computing: Randomized Algorithms and Probabilistic Analysis. Cambridge University Press, 2005.

Selbstständigkeitserklärung

Ich erkläre gegenüber der Freien Universität Berlin, dass ich die vorliegende Dissertation selbstständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt habe. Die vorliegende Arbeit ist frei von Plagiaten. Alle Ausführungen, die wörtlich oder inhaltlich aus anderen Schriften entnommen sind, habe ich als solche kenntlich gemacht. Diese Dissertation wurde in gleicher oder ähnlicher Form noch in keinem früheren Promotionsverfahren eingereicht. Mit einer Prüfung meiner Arbeit durch ein Plagiatsprüfungsprogramm erkläre ich mich einverstanden.

Datum:

Unterschrift:

Marvin Lücke