UNIVERSIDADE D
COIMBRA

Sérgio Manuel Carvas Machado

Learning the Graph of Networked Dynamical Systems under Partial-observability via Artificial Neural Networks

Dissertation in the context of the Master in Informatics Engineering, specialization in Intelligent Systems, advised by Dr. Augusto Santos and Professor Bernardete Ribeiro, and presented to the Department of Informatics Engineering of the Faculty of Sciences and Technology of the University of Coimbra.

September of 2022

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE D
COIMBRA

DEPARTMENT OF INFORMATICS ENGINEERING

Sérgio Manuel Carvas Machado

Learning the Graph of Networked Dynamical Systems under Partial-observability via Artificial Neural Networks

Dissertation in the context of the Master in Informatics Engineering, specialization in Intelligent Systems, advised by Dr. Augusto Santos and Professor Bernardete Ribeiro, and presented to the Department of Informatics Engineering of the Faculty of Sciences and Technology of the University of Coimbra.

September of 2022

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE D
COIMBRA

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

Sérgio Manuel Carvas Machado

Aprendizagem do Grafo de Sistemas Dinâmicos em Rede sob Observabilidade Parcial por Redes Neuronais Artificiais

Dissertação no âmbito do Mestrado em Engenharia Informática, especialização em Sistemas Inteligentes, orientada pelo Dr. Augusto Santos e Professora Bernardete Ribeiro, apresentada ao Departamento de Engenharia Informática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

Setembro de 2022

Acknowledgements

I would like to express my deepest gratitude to all those who took part in this research either in a more active or less present way.

To Dr. Augusto Santos for the enormous knowledge contribution, particularly on the areas of Networked Dynamical Systems and Graph Learning, along with his constant active presence and advice during this work. He had a major contribution to this research. During all phases of this research, he made sure I could improve at every level and showed to be an excellent mentor. To Professor Bernardete Ribeiro, who, in an active manner, expressed her interest in the addressed topics and explored processes, and provided numerous advice on how I could refine my work. On top of that I praise her concern that I was still motivated and the work was flowing properly. To Professor Catarina Silva who followed and advised the initial exploratory work. Also, to Anirudh Sridhar for the feedback and contribution to this work. Last but not least, to Professor José Moura whose enormous knowledge contributed to this work and its refinement, and for granting the possibility of continuing this research.

This endeavor would not have been possible without my parents and sisters' extraordinary support throughout my course, during the good and bad moments. Thank you for all the effort so that I didn't lack anything during these years. That effort was certainly not in vain.

With regard to personal and professional terms, the present research work was not only crucial to enhance my technical skills regarding the subjects of Deep Learning and Graph Learning, but also to strengthen my soft skills. This academic internship allowed me to put to the test my commitment and responsibility, to improve my communication ability, and most importantly, the development of critical thinking.

This work was partially funded by the FCT - Foundation for Science and Technology, Portugal, I.P./MCTES through national funds (PIDDAC), within the scope of CISUC I&D Unit with reference UIDB/00326/2020.

Abstract

In this thesis, we address the problem of graph identification of linear stochastic networked dynamical systems from the time series data that reflect the state evolution of a subset of observed nodes in the system – hence, a complementary subset of nodes lies unobserved or latent. Given these node-level time series data, the goal is to consistently recover the underlying graph of dependencies among the nodes in the networked system. We assume partial observability, i.e., the time series data of only a subset of nodes comprising the network is available. We propose a novel feature vector computed from the observed time series as a statistical descriptor for the coupling between nodes. We formally prove that these features are *linearly separable*, that is, there exists a hyperplane (in feature space) that partitions the set of features associated with connected pairs of nodes from those associated with disconnected pairs of nodes. This separability property allows the use of these features to train a multitude of classifiers in order to perform graph structure identification. In particular, we choose to train Convolutional Neural Networks (CNNs) over these features with a resulting graph learning algorithm that outperforms state-of-the-art counterparts w.r.t. sample-complexity, i.e., number of samples required to reach a certain level of accuracy in the recovery of the graph. While the CNNs are trained over a particular synthetic network, they generalize well over networked systems with distinct connectivity patterns (dense or sparse) including real-world networks, and distinct noise-level regimes. This is an important property as, in general, we might have no information about these structural attributes. Finally, the proposed method consistently learns the graph in a pairwise manner, that is, via inferring whether a particular pair of nodes is connected or not from their time series (ignoring the time series from other nodes). This is particularly tailored to large scale systems where observation of all nodes in the network is unfeasible.

Keywords

Graph Learning, Structure Identification, Artificial Neural Networks, Causal--Inference, Partial-Observability.

Resumo

Nesta tese, abordamos o problema da identificação de grafos de sistemas dinâmicos lineares estocásticos em rede, a partir de dados de séries temporais que refletem a evolução do estado de um subconjunto de nós observados no sistema. Assim sendo, um subconjunto complementar de nós permanece não observado ou latente. Dadas essas séries temporais ao nível do nó, o objetivo é recuperar consistentemente o grafo subjacente relativo às dependências entre os nós no sistema. Assumimos observabilidade parcial, isto é, apenas estão disponíveis um subconjunto das séries temporais dos nós que compõem a rede. Propomos um novo vetor de features calculadas a partir das séries temporais observadas, que agem como um descritor estatístico do acoplamento entre nós. Provamos formalmente que essas features são *linearmente separáveis*, ou seja, existe um hiperplano (no espaço de features) que separa o conjunto de features associadas a pares de nós conectados daquelas associados a pares de nós desconectados. Essa propriedade de separabilidade permite usar essas features para treinar um grande número de classificadores e efetuar a identificação da estrutura do grafo subjacente. Em particular, optamos por treinar Redes Neuronais Convolucionais (CNNs) sobre essas features resultando num modelo de aprendizagem que exibe uma performance, em geral, superior a outros algoritmos state-of-the-art. Esta performance refere-se à complexidade amostral, isto é, número de amostras da série temporal necessárias para atingir um certo nível de precisão na recuperação do grafo. Enquanto as CNNs são treinadas em um grafo sintético específico, elas generalizam bem para sistemas de rede com padrões de conectividade distintos (densa ou esparsa), incluindo grafos do mundo real e regimes distintos de nível de ruído. Esta é uma propriedade importante, pois, em geral, podemos não ter informações sobre esses atributos estruturais. Por fim, o método proposto aprende consistentemente o grafo de maneira par-a-par, ou seja, inferindo se um determinado par de nós está conectado ou não a partir de suas séries temporais (ignorando as séries temporais de outros nós). Isso é particularmente adequado para sistemas de grande escala, onde a observação de todos os nós da rede é inviável.

Palavras-Chave

Aprendizagem Automática de Grafos, Identificação de Estrutura, Redes Neuronais Artificiais, Inferência Causal, Observabilidade Parcial.

Contents

Acronyms

AB Agent-Based.

ABM Agent-Based Modeling.

ANNs Artificial Neural Networks.

CA Cellular Automata.

CI Conditional Independence.

CNNs Convolutional neural networks.

EEG Electroencephalogram.

FFNNs Feed-Forward Neural Networks.

FMRI Functional Magnetic Resonance Imaging.

GC Granger Causality.

GIS Geographic Information Systems.

GMM Gaussian Mixture Model.

LBP Local Binary Pattern.

MSE Mean Squared Error.

ODE Ordinary Differential Equation.

PCI Perturbation Cascade Inference.

SDE Stochastic Differential Equation.

SEIR Susceptible-Exposed-Infected-Susceptible.

SIFT Scale Invariant Feature Transform.

SIR Susceptible-Infected-Removed.

SIS Susceptible-Infected-Susceptible.

SVM Support Vector Machines.

List of Figures

Chapter 1

Introduction

The current chapter presents a brief motivation for the main scope of this thesis, namely, learning the graph of interactions underlying linear networked dynamical systems from the observed time series, describes the contributions of this work and sets the outline for the rest of the thesis. In particular, Section 1.1 presents the main motivation for the problem. Section 1.2 discusses our main goals and contributions. Section 1.3 lays down the outline for the rest of the thesis.

The results were submitted in part for publication and a preprint can be found at [Machado et al., 2022]. Other results are in preparation.

1.1 Motivation

In the last few years, mathematical models have been increasingly used to support the public health strategy concerning the mitigation of infectious diseases. Particularly, in developing response plans for pandemic outbreaks such as new strains of influenza A and, more recently, for the COVID-19 pandemic. An epidemiological model attempts to describe qualitatively the evolution of the fraction of infected individuals across distinct regions of the globe. They allow the study of spreading dynamics and simulating the effect of preventive measures, such as vaccination or quarantine.

Networked epidemic models play a critical role in describing the spreading of infectious diseases. Mobility of infected individuals across communities is one of the main causes of transmission. These contacts among communities are usually unknown (or not transparent), i.e., the underlying contact network topology is unknown. On the other hand, mitigation policies necessarily rely on information about this contact network. For instance, a possible framework to mitigate virus propagation is to quarantine a subset of communities (or nodes in the network) to maximally disconnect the contact network, a paradigm also known as network dismantling [Braunstein et al., 2016; Ren et al., 2019]. Clearly, this family of mitigation strategies rely on concrete knowledge about the underlying network of contacts.

Therefore, these causal relationships among nodes in the networked system need to be consistently inferred from the observed evolution of the pandemics – e.g., in the form of time-series reflecting the evolution of the number of infected individuals per community. For large-scale networks, the inference must be performed under partial observability: only a subset of the nodes in the network is observed. There are several reasons that cause the observability limitation in large-scale networks such as: (i) **[accessibility-limit]**, some parts of the network are inaccessible or some interaction sources are unknown; (ii) **[probing-limit]**, acquiring and storing data capacity may be smaller than the network scale; (iii) **[processing-limit]**, data-mining complexity may limit the size of data that can be processed [Santos et al., 2020a].

Learning the graph of interactions underlying networked dynamical systems (under partial observability) is an emergent topic with a lot of interest and many open questions. It finds applications in epidemics, specially in what concerns the design of mitigation policies, but it naturally extends to a broad class of networked dynamical systems such as brain activity, as recent evidences show that the connectivity pattern among distinct active regions of the brain conveys important information about several forms of motor activities or cognitive disorders [Lehnertz et al., 2020; Oltra et al., 2021]; or even in finance since the dynamics of stock prices can be affected by interactions between corporations [Bazzi et al., 2015]. The common element about these examples is that the state-evolution (or time series data) of some nodes in the networked system is often available, but the underlying contact network is unknown or not fully known, while critically impacting the evolution of these systems.

1.2 Research Goals

In this thesis, we focus on linear stochastic networked dynamical systems. We will assume that the time series of only some nodes in the networked system is observed or processed. We remark that the framework of partial observability is necessarily more challenging than the full observability counterpart as the time series (or state evolution) of the observed nodes is critically impacted by the unobserved nodes time series. Nevertheless, under this challenging framework, the main goal is to consistently infer the network structure from the observed time series data. Fig. 1.1 summarizes the paradigm.

The main contribution of this thesis can be summarized as follows: (i)**[New features]** we propose a novel set of feature vectors computed from the time series to characterize the connectivity between nodes; (ii)**[Separability]** this features are shown to have a special property: they are linearly separable, in that, there is a hyperplane (in feature space) that consistently stratifies the features stemming from connected pairs and those from disconnected pairs; (iii)**[Locality]** the *causal inference*, i.e., determining whether there is a link between two nodes is performed only via processing the time series of the two nodes (that is, via ignoring the time series of the rest of the network); (iv)**[CNN-approach]** with the referred separability property, we can resort to distinct machine learning tools

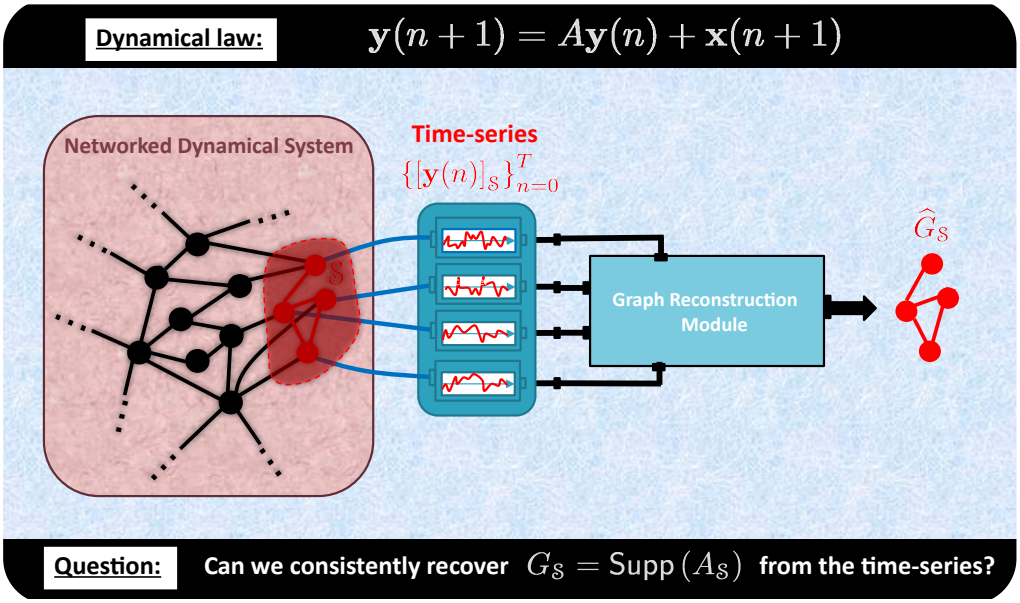


Figure 1.1: Graph structure identification under partial observability. The goal is to devise a mechanism that consistently determines the subgraph structure linking the subset of observed nodes $\mathcal{S}$ from the corresponding observed time-series. These time-series reflect the state-evolution of the observed subset of nodes $\mathcal{S}$ of the networked dynamical system. For simplicity, we illustrate an undirected graph; however, the paradigm proposed in this thesis successfully applies to directed graphs. The subgraph of interest is the support of the interaction submatrix $A_{\mathcal{S}}$ supporting the contacts among the observed nodes $\mathcal{S}$.

in order to perform inference and in this thesis we choose to train convolutional neural networks with a resulting algorithm that exhibits competitive state-of-the-art performance.

1.3 Outline

Next, we provide an overview of the content of this thesis.

- **Chapter 2** describes the main theoretical background components underlying the research that was carried out. We briefly discuss on networked dynamical systems (using primarily epidemiological models as reference); on network generative models; on Artificial Neural Networks and on the paradigm of causal inference or graph learning of networked dynamical systems (possibly under partial observability);

- **Chapter 3** presents a concise discussion of related works on graphical models and dynamical systems. We divide the methods present in the literature into two distinct groups, depending on whether it is capable of recovering the structural connectivity of a graph under full-observability or partial-observability.

- **Chapter 4** presents the problem formulation and main theoretical results that motivate the proposed CNN-based approach for graph learning, in particular, it is shown that the proposed set of features is linearly separable. Further, we introduce the methodology employed to perform the numerical experiments exhibited on the next chapter;
- **Chapter 5** validates the consistency of the method and demonstrates via several numerical experiments the overall superiority of the approach as compared with other state-of-the-art methods. In particular, it shows that our CNN-based method is quite robust across networked dynamical systems with distinct connectivity patterns (densely or sparsely connected);
- **Chapter 6** presents final remarks and offers distinct open directions for future research.

Chapter 2

Background

This chapter entails some of the background that is relevant for the rest of this thesis. Section 2.1 briefly introduces the main elements of dynamical systems theory. In Section 2.2, we present deterministic and stochastic epidemiological models as they represent important examples of networked dynamical systems. Section 2.3 briefly describes some basic concepts regarding Artificial Neural Networks (ANNs) and Convolutional neural networks (CNNs). In Section 2.4, we briefly discuss some works on parametric inference over epidemiological models via ANNs. Finally, the last sections introduce some concepts of our main research topic: Section 2.5 shows an approach to causal inference or graph learning under full observability, Section 2.6 focus on partial observability and Section 2.7 discusses on the general ill-posed nature of graph learning.

2.1 Dynamical Systems

Dynamical systems theory is a direct off-spring of *calculus* developed by Isaac Newton and Wilhelm von Leibniz in their pursuit to comprehend celestial mechanics, namely, in mathematically describing the evolution over time of the position of celestial bodies. For the most part, the great breakthrough from Newton and Leibniz (that ripples to this day) was to compactly describe a (dynamical) law whereby all complexity emerges from, namely, all motion ticks forward with the *arrow of time*. The law was in the form of an Ordinary Differential Equation (ODE) and to describe the evolution of the system meant to find the solutions to these ODEs. From the XVII century up until early the XX century, the theory revolved about this particular philosophy, i.e., finding the closed-form solutions to these ODEs. The modern theory of dynamical systems was born with Henri Poincaré after realizing that not only this is not, in general, possible, but sometimes it is not helpful – as the expression for the solution could be too complex. The modern theory revolves around more *qualitative* descriptors of the system: what does the long-term behavior of a dynamical system look like? Does the system slow down to an equilibrium? Loops indefinitely (limit cycle)? Does it accumulate onto some compact region of the state-space (attractor)? It is further in light of this modern view that ODEs yield standard models for natural phenomena: the

idea is that some elements of the qualitative behavior of the ODE models match observed phenomena – only in rare cases, ODEs provide a precise quantitative description of nature.

Networked Dynamical Systems represent a subclass of dynamical systems modeling the state evolution of the nodes in a network. In this framework, the state of the nodes evolves due to their neighboring interactions, i.e., respecting the underlying sparsity imposed by the network. An example of a networked dynamical system is a pandemic where: (i) **nodes** can be cast as communities (e.g., cities, countries, etc.); (ii) **connections** abstract the possible *avenues of interaction* among these communities that foster the spread of a virus; (iii) **the state** is given by the fraction of infected individuals over time. Another example is brain activity where each node may represent a region of the brain and the state evolution in the form of time series data can be captured by the Functional Magnetic Resonance Imaging (FMRI) or Electroencephalogram (EEG) signals. An important remark is that the topology of the underlying network plays an important role in the qualitative behavior of these systems. While this is the case, information about the actual network is often not available directly, but lurks in the observed time-series associated with the state-evolution of the system. In these settings, the main goal of this thesis is to resort to ANNs in order to characterize consistently this causal information from the observed time-series.

2.1.1 Networked Dynamical System

Continuous-time networked dynamical systems can be cast as the solution to (possibly stochastic) ODEs. An ODE is an equation of the form

$$\dot{\mathbf{x}}(t) = \mathbf{F}\left(\mathbf{x}(t)\right), \tag{2.1}$$

where $\mathbf{F} : U \subset \mathbb{R}^N \longrightarrow \mathbb{R}^N$ is the so-called vector-field. A solution to (2.1) is a curve $\mathbf{x} : \mathbb{R}_+ \longrightarrow \mathbb{R}^N$ fulfilling the identity (2.1) where $\mathbb{R}^N$ (or an open subset thereof) is the state-space (the set of all possible states of the system) [Alligood et al., 2000]. Therefore, a curve x is solution to (2.1) whenever at each point $\mathbf{x}(t) \in U$, its tangent vector is given by $\mathbf{F}(\mathbf{x}(t))$. Fig. 2.1 graphically illustrates the idea.

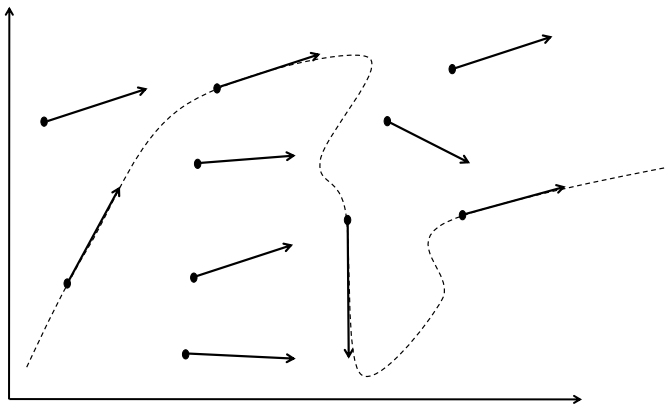


Figure 2.1: Example of a trajectory *fitting* the vector-field, i.e., of the solution of an ODE.

In general, a closed-form solution to the ODE (2.1) cannot be drawn [Strogatz, 1994] and numerical methods are the principal tools to integrate (2.1) and characterize solutions. This subject is also known as numerical integration and it is described in more detail in Section 2.1.4.

If we model the solution $\mathbf{x}(t) = (x_1(t), x_2(t), \ldots, x_N(t))$ as the state of the N nodes in a network at time t, then, we say that the node j affects (directly) the node i whenever the vector-field at the node i $F_i(x_1, x_2, \ldots, x_N)$ is sensitive to its jth entry. This implies that $\dot{x}_i(t)$ is sensitive to the state $x_j(t)$. More compactly, this is equivalently to saying that the Jacobian matrix $DF(x)$ entails the network structure of interactions in its support:

$$[DF(x)]_{ij} \neq 0 \Leftrightarrow j \to i. \tag{2.2}$$

When the underlying network structure exhibits nontrivial sparsity, we refer to (2.1) as a networked dynamical system. An example is the linear ODE

$$\dot{x}(t) = Ax(t) \tag{2.3}$$

where the support of the matrix A conveys the network structure information. In the causal identification framework, the main goal is to identify A or the support of A from observation of the time-series $\{\mathbf{x}(t_n)\}_{n=1}^{T}$.

2.1.2 Properties of Dynamical Systems and ODEs

In its utmost generality, a dynamical system is a family of maps $\phi_t : X \longrightarrow X$ from a state-space X onto itself and indexed by *time* $t \in \mathbb{T}$ [Ott, 2002]. $\phi_t(x)$ represents the state of the system at time t provided that its initial state was $x \in X$ and fulfils the following properties: (i)**[identity]** $\phi_0(x) = x$ for all $x \in X$; (ii)**[semigroup]** $\phi_{s+t}(x) = \phi_t(\phi_s(x))$. The dynamical system ϕ is often referred to as a *flow* and it is continuous-time (respectively, a discrete-time) dynamical system if the cardinality of $\mathbb{T}$ is that of $\mathbb{R}$ (respectively, is that of $\mathbb{N}$).

More concretely, solutions to ODEs

$$\dot{\mathbf{x}}(t) = F(\mathbf{x}(t)) \tag{2.4}$$

are necessarily flows $\phi_t(x)$, i.e., fulfill the aforementioned group properties, under certain regularity assumptions on F. The space where all possible states of a system are represented is called state space. The zeros of the vector-field F define the equilibria of the system, i.e., the points $x^\star \in \mathbb{R}^d$ so that $\mathbf{x}(t) = \mathbf{x}^\star$. An equilibrium $\mathbf{x}^\star$ may be locally stable or unstable depending on whether small perturbations about $\mathbf{x}^\star$ yield convergence to $\mathbf{x}^\star$ or not. Fig. 2.2 shows an example for the 1D ODE $\dot{x} = x^2 - 1$, where $x^\star = -1$ represents a globally stable equilibrium and $x^\star = 1$ represents an unstable equilibrium.

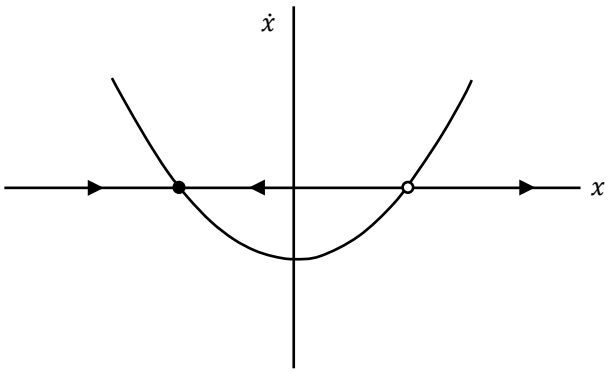


Figure 2.2: Example of a phase portrait for the ODE $\dot{x} = x_2 - 1$.

Definition 2.1 (Lyapunov stability)**.** *An equilibrium x^* is Lyapunov stable if for every $\epsilon > 0$ there exists a $\delta > 0$ such that if $|x - x^*| < \delta$ then*

$$|x(t) - x^*| \text{ for all } t \geq 0.$$

The equilibrium is asymptotically stable if it is Lyapunov stable and there exists $\eta > 0$ such that if $|x - x^*| < \eta$ then

$$x(t) \to x^* \text{ as } t \to \infty.$$

In other words, stability of a trajectory indicates that if it starts close to the equilibrium, stay arbitrarily close for $t \geq 0$. The concept of asymptotic stability, means that nearby trajectories also approach the equilibrium as $t \to \infty$. Lyapunov stability does not suggest asymptotic stability as nearby solutions could oscillate around an equilibrium without converge towards it. Additionally, near solutions approaching the equilibrium does not imply asymptotic stability since trajectories could take extensive paths before reaching the equilibrium and this would violate Lyapunov stability [Hunter, 2011].

In the qualitative theory of dynamical systems, we are particularly interested in the long-term faith of the system. This naturally leads to the notion of attractor: a compact region in state-space where a *large* collection of trajectories of the dynamical system accumulates onto. To each attractor $\mathcal{A} \subset \mathbb{R}^d$, we can associate the set of all initial conditions $\mathcal{B}(\mathcal{A}) = \left\{\mathbf{x}_0 \in \mathbb{R}^d : \lim_{t\to\infty} d\left(\phi_t\left(\mathbf{x}_0\right), \mathcal{A}\right) = 0\right\}$ that lead the dynamical system to converge to $\mathcal{A}$, where we have defined

$$d\left(\mathbf{x}_0, \mathcal{A}\right) \triangleq \inf_{\mathbf{y}\in\mathcal{A}} ||\mathbf{x}_0 - \mathbf{y}||_2 . \tag{2.5}$$

The set $\mathcal{B}(\mathcal{A})$ is called the basin of attraction of $\mathcal{A}$ [Sayama, 2019]. Fig. 2.3 shows two examples of attractors: (a) the attractor is the origin of the state-space; (b) the attractor is the closed dashed curve.

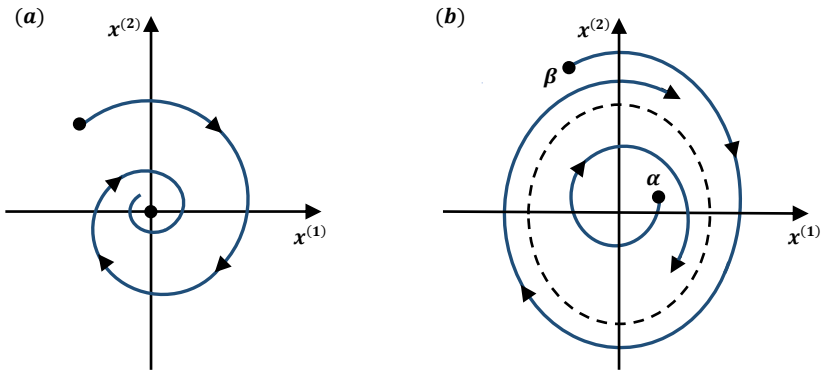


Figure 2.3: Example of attractors [Ott, 2002].

A critical property of dynamical systems is uniqueness: to each initial condition $\mathbf{x}_0 \in \mathbb{R}^d$, there is only one solution departing from $\mathbf{x}_0$. In other words, if $\mathbf{x}_0 \neq \mathbf{y}_0$, then $\phi_t(\mathbf{x}_0) \neq \phi_t(y_0)$ for all $t \geq 0$.

Two main forms of violation of this condition are: (i)**[coalescence]** trajectories coalesce eventually, i.e., $\phi_t(\mathbf{x}_0) = \phi_t(\mathbf{y}_0)$ for some $t > 0$; (ii)**[bifurcation]** a trajectory splits in two eventually. The former implies *loss of information*: given the current state, we cannot infer the initial states. In particular, one of the main foundational assumptions of physics (conservation of information) is not consistent with (i). On the other hand, bifurcation violates *determinism*: given an initial state, it is not possible to assert its future.

To sum up, the uniqueness property is foundational for dynamical systems and its off-spring applications. This condition is met, for instance, whenever the vector-field fulfills some mild regularity assumptions, e.g., it is Lipschitz continuous (Picard-Lindelöf Theorem).

Definition 2.2 (Lipschitz continuity)**.** *Let U be an open set in $\mathbb{R}^d$. A function $F: \mathbb{R}^d \longrightarrow \mathbb{R}^d$ is Lipschitz continuous over U, if there is a constant $L > 0$ such that*

$$|F(x) - F(y)| \leqslant L|x - y|$$

for every $x, y \in U$. The constant L is called the Lipschitz constant of F.

Examples of Lipschitz continuous functions are the continuously differentiable functions, i.e., the set of functions whose partial derivatives exist and are continuous [Alligood et al., 2000].

Finally, we must refer to a specific class of dynamical systems called monotonous or order-preserving, that can be found in many biological, physical and economical dynamical models [Hirsch and Smith, 2006]. Namely, a monotonous flow is characterized by the following property

$$u \leq v \ \ and \ \ t \geq 0 \implies \phi_t(u) \leq \phi_t(v) \tag{2.6}$$

In other words, the flow map $\phi_t : \mathbb{R}^d \longrightarrow \mathbb{R}^d$ preserves the vector order '$\leq$'. Such monotonous dynamical models exhibit certain restricted long-term behavior. For

instance, in $\mathbb{R}^d$ with the standard vector ordering, there cannot be stable periodic orbits other than equilibrium. Moreover, the equilibrium of the system is also typically a *global attractor*: (almost) all trajectories of the system converge to a point in the equilibrium.

2.1.3 Considerations on Continuous-Time Dynamical Systems

Generally ODEs that model the transmission of infectious diseases are described for being continuous in time and having a nonlinear evaluation, as stated in the following sections. Although, under certain conditions it is possible to represent a nonlinear dynamic system by a similar and simpler linear system. Also, we can discretize a dynamical system continuous in time. In this section we present two concepts of dynamic systems that allow this transformation, and endorse the approach taken in this thesis.

Simplification of nonlinear dynamics. The Hartman-Grobman theorem allows the representation of the local phase portrait about certain types of *equilibrium* in a nonlinear system by a similar and simpler linear system. This is achievable by computing the Jacobian matrix of the system at the equilibrium point. Consider following the differential equation

$$\dot{x} = F(x(t)) \tag{2.7}$$

where $F \in C^1(U, \mathbb{R}^n)$ and U is an open subset of $\mathbb{R}^n$. Assume that $x^* \in U$ is an equilibrium, that is $f(x^*) = 0$. The linearized dynamic system associated to 2.7 near x^* is

$$\dot{x}(t) = A(x(t)) - Ax^* \tag{2.8}$$

where $A = DF(x^*)$ is the derivative of F at x^*. The equilibrium x^* is said to be hyperbolic if the matrix A has no purely imaginary eigenvalue. Dynamical systems 2.7 and 2.8 are designated topologically conjugate at x^* if there are neighborhoods X, Y of x^* in U and a homeomorphism (continuous bijection with a continuous inverse) h: $X \to Y$ mapping the orbits of 2.7 in X into the orbits of 2.7 in Y in a time-preserving way [Baratchart et al., 2006].

This theorem is particularly compelling to us, not only because it bears a way of simplifying the underlying dynamics, but also because the Jacobian matrix contains the causal information for networked dynamical systems. Suppose we are observation the nodes i, j of a certain network, the node j has an effect on a node i whenever $\dot{x}_i$ depends of the entrance $\dot{x}_j$ in $\dot{x}_i(t) = F(x_1(t), x_2(t), x_3(t), ...)$, i.e. j as direct impact on i.

Continuous-time discretization. The previous sections focused on continuous-time dynamical systems, however discrete-time systems may naturally arise from the continuous ones. Consider the following Stochastic Differential Equation (SDE)

$$dx(t) = Ax(t)dt + db(t). \tag{2.9}$$

It can be shown that the time-series samples with sample interval T obeys the law

$$\mathbf{x}(n+1) = e^{AT}\mathbf{x}(x) + \mathbf{w}(n+1), \tag{2.10}$$

and if T is *small* enough, then the previous law reduces approximately to

$$\mathbf{x}(n+1) \approx (I + AT)\mathbf{x}(x) + \mathbf{w}(n+1), \tag{2.11}$$

where the term AT conveys the underlying ground-truth graph of interactions linking the nodes in the network and $\{\mathbf{w}(n)\}_{t\geq 0}$ is a sequence of i.i.d. normal vectors. In other words, devising inference tools for discrete-time models (which will be the focus of this thesis) is key for a great class of continuous-time systems. A similar discretization approach is adopted in [Bento et al., 2010].

2.1.4 Ordinary Differential Equations Solvers

We might be unable to find the solution to general ODEs in closed form. Numerical methods, provided by ODE solvers, are important tools to study these systems quantitatively. The fundamental idea behind ODE solvers is discretizing time to obtain a discrete-time system counterpart whose evolution lies as close as possible to the continuous system. The smaller the step size the better the approximation of the ODE solution [Press et al., 2007]. Euler's method represents one of the most basic forms of discretization: define the sequence of time instants $t_{n+1} = t_n + h$ and rewrite the ODE $\dot{x}(t) = F(x(t))$ as

$$\frac{x(t_{n+1}) - x(t_n)}{h} = F(x(t_n)). \tag{2.12}$$

Nonetheless, this method is not *optimal* in the sense that it is prone to numerical instabilities and only uses derivatives at the beginning of the interval so it's not very accurate compared to other methods with the same step-size h. For it to be accurate we would have to use a very small step size.

The accuracy of a numerical integration method algorithm is dependent on the round-off error that arises from the maximal round-off error in the computer number representation, by the product of the integration interval. The truncation error results from the true integration value subject to a constant round-off error. This error can be decreased by reducing the step-size which reduces the truncation error and by using a higher-order integration formula.

A higher-order method takes into account more steps between the integration interval in order to improve accuracy. Euler's method is very asymmetric regarding the beginning and end of the interval. For a second-order method, taking an exploratory step to the midpoint of the interval and then using the value of that point to compute the actual value of the whole interval yields a more symmetric integration method. This is the idea behind Runge-Kutta method, the symmetrization cancels out the first-order error. Therefore, we should not stop in a second-order method and so we should consider taking more trial steps. The limiting factor when numerically integrating is the computational effort when

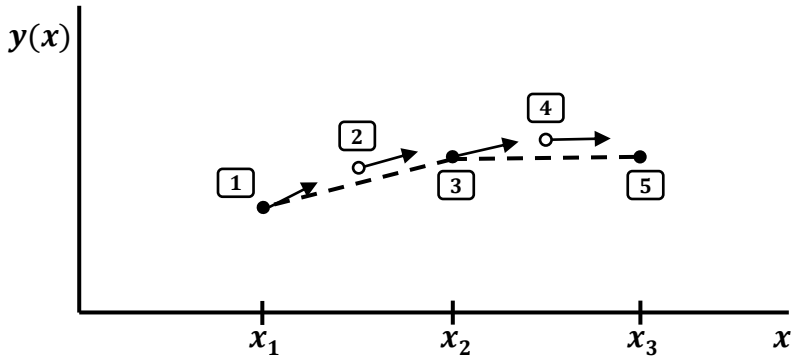


Figure 2.4: Illustration of the second-order Runge-Kutta.

computing the ODE equation. Since the Runge-Kutta method considers n evaluations where n is the integration order, we must find a trade-off between a low computation effort and truncation error per step h. The most popular method is the fourth-order Runge-Kutta which considers four evaluations per each step h – this is the numerical integration method that we adopt in this thesis.

$$\begin{aligned} k_1 &= hf(t_n, y_n) \\ k_2 &= hf(t_n + \frac{h}{2}, y_n + \frac{k_1}{2}) \\ k_3 &= hf(t_n + \frac{h}{2}, y_n + \frac{k_2}{2}) \\ k_4 &= hf(t_n + h, y_n + k_3) \\ y_{n+1} &= y_n + \frac{k_1}{6} + \frac{k_2}{3} + \frac{k_3}{3} + \frac{k_4}{6} \end{aligned} \tag{2.13}$$

This method can be further improved with an adaptive step size algorithm in order to achieve better accuracy with less computational cost. Nonetheless, this *adaptive* idea cannot be used for the work of this thesis since we want to know the ODE solution for a given time point so a fixed step size method is required.

Higher order Runge-Kutta methods are suitable for problems where moderate accuracy is needed and computational efficiency is not an issue. For problems where high accuracy is required, we should use other methods such as Richardson extrapolation or Predictor-corrector. Richardson extrapolation is based on the idea of extrapolating a computed solution value, to the one that should have been attained if the step size was way smaller than what it really is. Extrapolating a solution refers to the use of known values to project a value outside of the intended range of those values. Furthermore, the Predictor-corrector method stores the computed results and consequently uses those results to extrapolate the solution one step ahead and so it has a comparatively big overhead [Press et al., 2007].

2.2 Modeling Infectious Diseases

The transmission of infectious diseases through interactions in a population is very complex, which makes it difficult to understand the large-scale dynamics of

disease spread without the formal structure of a mathematical model. An epidemiological model makes use of the behavior of an infectious individual (microscopic level) to predict the role of the spreading of a disease throughout the population (macroscopic level). These models, disregarding type or complexity, are essentially simplifications of a real-life system, which may contain only some of the fundamental elements of it as defined by the researcher [Sattenspiel, 2002].

The mathematical modeling of infectious disease is advantageous to test theories about the spread of the infection such as comparing different diseases in the same population or vice-versa [London and Yorke, 1973]. Epidemiological methods are also useful to compare the impact of prevention and control procedures such as in the work by [Hethcote and Yorke, 1984] that uses models to compare different control programs for gonorrhea.

In the following subsections we will describe three distinct approaches to model the spread of infectious diseases. The first approach, called compartmental models, are deterministic and describe the *macroscopic* level of a disease spread. The second method is stochastic and makes use of network models to describe the *microscopic* level of the spread. The third technique, based on the complex systems theory, is a bottom-up approach (describes the *microscopic* level) which incorporates the spatial aspects of the spread of an epidemic. These refereed methods, in contrast to others such as statistical models that learn patterns from the historical time series, allow the representation of the process of infection between two individuals and, consequently, mathematically describe the evolution of an epidemics over time of a whole population.

2.2.1 Epidemiological Compartmental Models

The idea behind compartmental modeling was introduced by [Kermack and McKendrick, 1927] as the first ODE-based rigorous framework for modeling infectious diseases as it is known today as the Susceptible-Infected-Removed (SIR) model [Bacaër, 2011].

In this model, the studied population is assigned to one of four classes: S represents the fraction of the population that is susceptible, I stands for the fraction of infected individuals and R represents the fraction of recovered individuals. Within this paradigm, distinct models can be drawn such as Susceptible-Infected-Susceptible (SIS), SEIR, SEIRS and others [Kretzschmar, 2016].

We would use the SIR terminology to model an infectious disease that offers immunity to individuals against reinfection and a SIS terminology for a disease that does not confer immunity against reinfection. Individuals flow from the susceptible class, to the infected and back to the susceptible class.

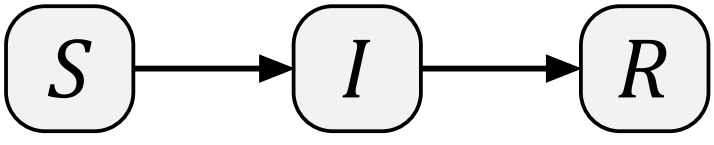


Figure 2.5: Flow chart of the SIR model.

The fundamental element of these models is the rate that expresses the transmission of the infection according to a mass action incidence, that is, assuming that the individuals of the population meet each other randomly with the same probability per time unit. The compartmental model 2.5 is translated into a system of ODEs

$$\begin{aligned} \frac{dS}{dt} &= -\beta SI \\ \frac{dI}{dt} &= \beta SI - \gamma I. \\ \frac{dR}{dt} &= \gamma I \end{aligned} \tag{2.14}$$

The parameters β and γ are the infection and recovery rate respectively. β depends on the number of contacts per unit time κ and the transmission rate τ. Therefore, $\beta = \frac{\kappa \cdot \tau}{N}$ and $\gamma = \frac{1}{D}$, where N is the number of individuals and D the infection duration.

An essential element for predicting the evolution of an epidemic is how many people are infected by an infectious individual for the time he is infected and able to transmit the disease. That is R_0 the *basic reproduction number*, derived from R, the *reproductive rate* [Anderson and May, 1982], which is a good indicator of the near tendency of the epidemic. If $R_0 > 1$, I exponentially grows; if $R_0 < 1$, I exponentially decreases.

$$R_0 = \frac{\kappa \cdot \tau}{\gamma} = D \cdot \kappa \cdot \tau. \tag{2.15}$$

More complex models [He et al., 2020; Mateus et al., 2018] arise from the extension of the SIR model, namely, by considering additional states or compartments as illustrated in Figure 2.6. Examples of such compartments are the treated (T), vaccinated (V), quarantined (Q), hospitalized (H), the deceased (D), etc. Each transition from one compartment to another requires a rate. Moreover, to make the models more realistic [Carcione et al., 2020; Vaz and Torres, 2021] other parameters can be added such as the birth rate and nature and disease related deceased. Other types of extensions of the basic model [Rahmadani and Lee, 2020b] may consider the existence of multiple strains where different species interact among them.

Compartmental models have been commonly applied for different infectious diseases for a long time [Brauer, 2017]. These models are not very demanding in terms of the amount of input to be implemented. In fact they are favored in some situations such as diseases with a non-negligible rate of fatality. However, compartmental models have a downside as they assume that the epidemic process is deterministic, once the behavior of the infection is entirely determined by its history and the model rules and that and that the number of individuals belonging to a compartment is a time-differentiable function. Also, they presuppose that the population being studied is uniform and homogeneously mixing. Therefore, this approach may not be valid at the beginning of the outbreak when there are only a few infected and the transmission behavior is not clear [Brauer and Castillo-Chavez, 2012].

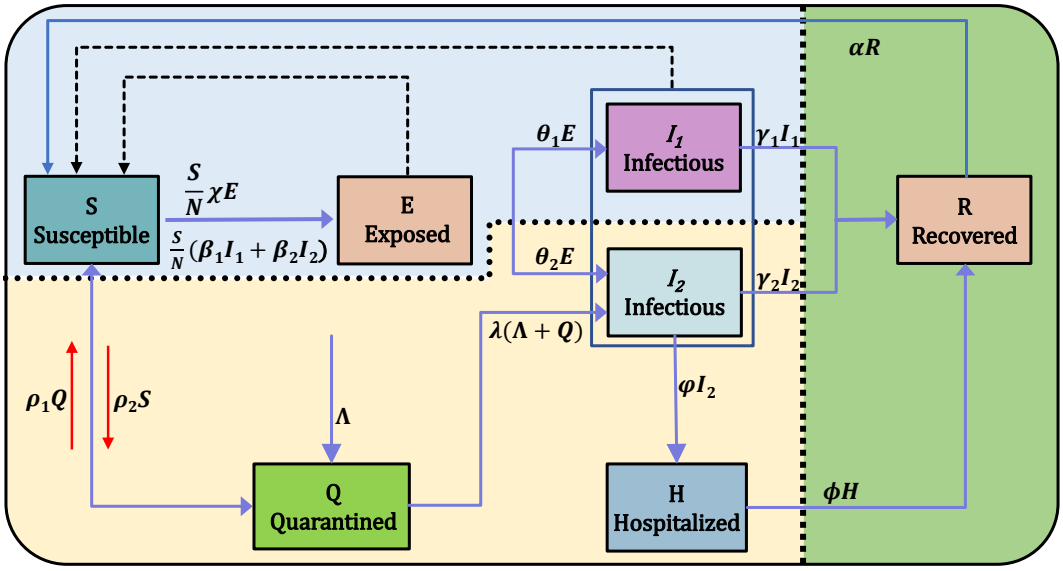


Figure 2.6: Flowchart of an extended SEIR model for COVID-19 modeling [He et al., 2020].

2.2.2 Epidemics over Networks

In the previous subsection we described models tailored to study the evolution of the pandemics over a single population. More general ODE-based compartmental models consider instead the spread across distinct communities. The dynamics of the SIS model over multiple communities can be given by the standard ODE model

$$\dot{y}_i(t) = \left(\sum_{j=1}^{N} A_{ij} y_j(t) \right) (1 - y_i(t)) - \mu_i y_i(t), \tag{2.16}$$

where $y_i(t)$ models the fraction of infected individuals at the community i at time $t \geq 0$; A_{ij} represents the *rate* that individuals from the community j contact with individuals in community i; and μ_i represents the healing rate at the community i. The term $(1 - y_i(t))$ entails the fraction of healthy (and thus, prone to be infected) individuals at the community i.

Equation (2.16) can be written in vector form as

$$\dot{\mathbf{y}}(t) = (A\mathbf{y}(t)) \odot (\mathbf{1} - \mathbf{y}(t)) - \mu \mathbf{y}(t), \tag{2.17}$$

where $A \in \mathbb{R}_{+}^{N \times N}$ is the matrix of interactions, i.e., the matrix conveying the rates of infection across communities; $\odot$ is the pointwise product between two vectors. Define $\overline{A}_{ij} = A_{ij}$ whenever $i \neq j$; $\overline{A}_{ii} = A_{ii} - \mu_i$; and $s(\overline{A}) = \max_{i=1,\ldots,N} \mathrm{Re}\lambda_i\left(\overline{A}\right)$ as the maximum real part among the eigenvalues of $\overline{A}$.

This model was first rigorously studied in [Lajmanovich and Yorke, 1976]. Some basic properties of this model established in [Lajmanovich and Yorke, 1976] are: (i) $0 \leq y_i(t) \leq 1$ for all $i = 1, 2, \ldots, N$ and all $t \geq 0$; (ii) the system $\mathbf{y}(t)$ reaches equilibrium regardless of the initial condition; (iii) If $s(A) > 0$ then the limiting equilibrium of the system is endemic regardless of the initial conditions,

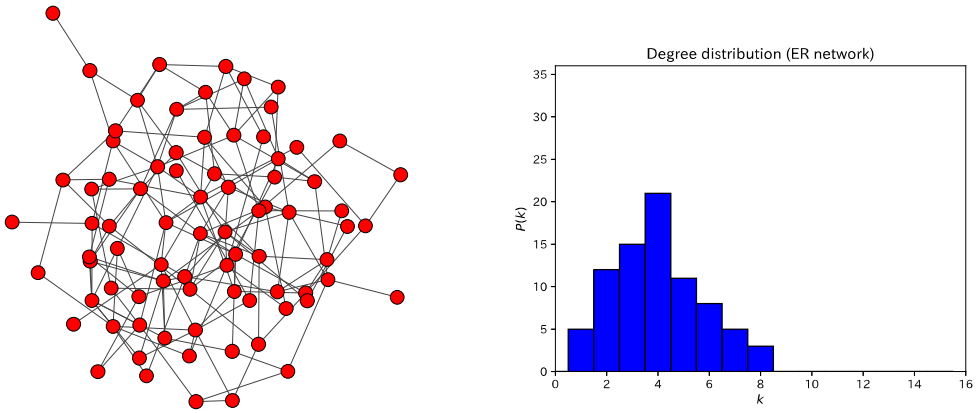
i.e., $\lim_{t\to\infty} \mathbf{y}(t) = \mathbf{y}^\star > \mathbf{0}$ for all initial conditions $\mathbf{y}(0) \in [0,1]^N$, otherwise, $\lim_{t\to\infty} \mathbf{y}(t) = \mathbf{0}$.

Point (i) states that $y_i(t)$ can be cast as the fraction of infected individuals (as it remains within $[0,1]$ for all time); (ii) states that the dynamical system neither oscillates (there are no limit cycles) nor it exhibits chaotic behavior: the system stabilizes after a transitory; (iii) says that the matrix of interactions A (along with the healing rates) determine the long-term faith of the system, namely, whether the virus persists or dies out. This latter point seems quite universal to networked dynamical systems in general: the network of interactions strongly determines the behavior of the system.

Network Generation

A network can be used to represent connections among individuals. They consist of nodes that may represent individuals, communities, devices, etc. Two nodes sharing the same edge are considered to directly interact and are referred to as neighbors. The number of neighbors of a node is specified as the degree D of that node. Here we present different types of network models. Note that, unless mentioned otherwise, we refer to networks as unweighted, undirected and with no selfloops or multi-edges.

The first well-known random network model was proposed by Erdös-Rényi. It assumes that edges connecting pairs of nodes are drawn independently with a probability p. For a network on N nodes, we have that Np is the mean degree of the nodes. In particular, each node has $n-1$ possible connections, and the number of neighbors follows a binomial distribution $D \sim Bin(n-1,p)$. As $n \to \infty$, this distribution approximates the Poisson distribution $D \approx P(p)$. Fig. 2.7 illustrates this network model.



(a) (b)

Figure 2.7: Erdös-Rényi Network, $N = 80$, $\langle\lambda\rangle = 4$ [Okabe and Shudo, 2021].

The Erdös-Rényi random model might not be ideal to capture the structure of certain real-world heterogeneous networks. In particular, it does not display community structure (with high probability) and its degree distribution is flat for the most part.

A second well-known network proposed by Watts and Strogatz [Watts and Strogatz, 1998] known as small world model, tries to produce more realistic networks. The process is based on choosing a node and an edge that connects it to its nearest neighbor. Then with a probability p, we reconnect the current ring to another chosen uniformly over the entire ring. The process is repeated while moving clockwise around the ring.

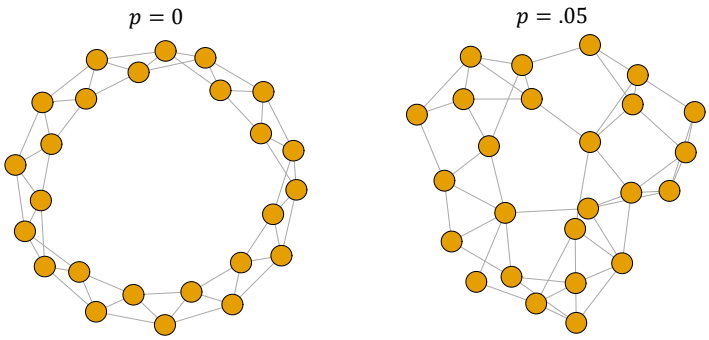
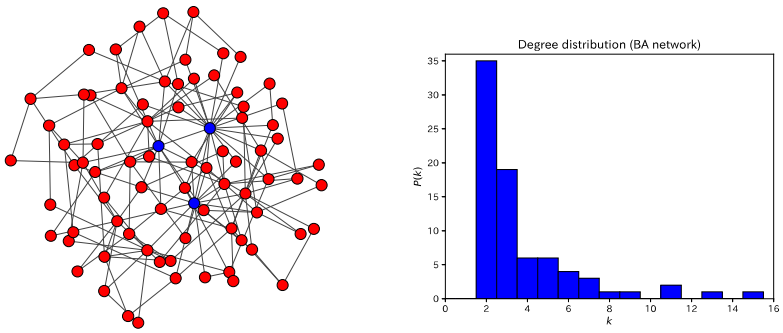


Figure 2.8: Small world networks with different probabilities.

The previous two models are still limited as they generate networks with degree distributions that do not describe real-world problems. For sexually transmittable diseases some individuals have many sexual partners while the rest have a small number of sexual partners. These are called scale-free networks once they have heavy-tailed degree distributions that approximately follow a power law [Luke, 2015].

The model proposed by Barabási-Albert [Barabasi and Albert, 1999] lies on this idea and follows an iterative growth concept based on preferential attachment; this means that a few nodes will have a large degree while the rest bear small degree. The building model starts with a single node without edges and, at each step, edges are added to the network. Also the probability of connecting to a particular node is equal to its current degree D, so there's a tendency to connect to a node with high degree which creates the preferential property. The randomness of this model arises from how the edges are connected to the existing nodes. Fig. 2.9 (a) illustrates a Barabási-Albert network of size $N = 80$ and the average connecting number is $\langle\lambda\rangle = 4$. The histogram (b) shows the distribution of the nodes degree.



(a) (b)

Figure 2.9: Barabási-Albert Network, $N = 80$, $\langle\lambda\rangle = 4$ [Okabe and Shudo, 2021].

Finally, there are models for dynamic networks where the current state of the epidemic affects the network morphology. The one proposed by [Britton et al., 2011], nodes may give birth to new nodes until they die and each node is equipped with a social index given at birth. During its life period a node creates edges to other nodes, higher social indexes perform this process at a higher rate and also edges disappear randomly in time.

Other dynamic models may reflect the state of the ongoing epidemic on the network. [Leung et al., 2018] presented a model where susceptible individuals may distance themselves from infected contacts as the epidemic evolves. Also, individuals may even replace lost social contacts by searching for new connections.

More on Spreading Models

Previously, we have described some ODE based compartmental epidemic models along with various methods to generate random networks. In this subsection, we describe two stochastic epidemic models (not ODE based).

The Reed-Frost Model [Britton, 2018] assumes that the infectivity of each individual is constant during the outbreak. It is a discrete-time dynamical process as infections happen through generations. At the first generation $k = 0$ one or more randomly selected individuals become infected, while the remainder are susceptible. In the next generations, infectious individuals spread the infection through its network neighbors independently, with a probability of p and then recover. Immune individuals or infected contacts are not affected. Those who became infected at generation k will become infectious at generation $k + 1$ and this continues until no new infections occur.

The general stochastic epidemic [Nåsell, 2002], often called Markov epidemic, assumes that individuals infect their neighbors independently at a rate γ and recover at a different rate μ. At time $t = 0$, one randomly selected individual is made infectious while the left becomes susceptible. Infectious individuals have independent contacts with their susceptible neighbors on the network randomly according to the Poisson distribution with a rate β. Also each Infected (I) person remains in that state for a period exponentially distributed with mean $1/\gamma$, i.e., $I \sim Exp(\gamma)$ until it recovers and develops immunity. Similarly to the Reed-frost model the epidemic continues until no new infection occurs.

As described, the Reed-Frost model takes place in discrete time, through generations, while the general stochastic epidemic model occurs in continuous time. However we could define the Reed-Frost model for a continuous time since it is how the infection outbreak happens in reality. Still, the concept of generations is a reasonable approach when there is a long period between exposure to the infection, followed by a small infectious period.

In both models there is only the possibility for an individual to infect its neighbors on the network which reflects the social proximity entailed by the graph. The major difference between these two epidemiological models lies in the infection event. For Reed-Frost models, distinct individuals become infected with inde-

pendent probabilities, while in general stochastic epidemiological models [Britton, 2020] these events incur nontrivial correlation. The distribution of the infected individuals on the network denotes the final outbreak, while the final size is the number of individuals that got infected during the outbreak, illustrated at [Britton, 2020].

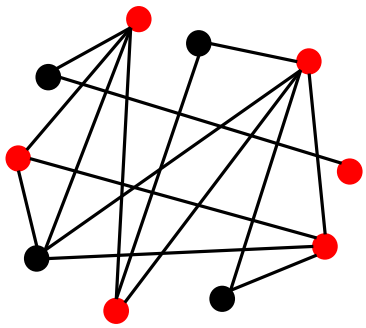


Figure 2.10: Illustration of the infection's final size in a random social network.

2.2.3 Complex Systems Approach

Differential equations models, presented before, neglect space implications within the systems and do not contemplate spatial and temporal factors such as variable population or dynamics. The disregard of the spatial element in the conception of epidemic models can be solved by using complex systems theory to address the spatial behavior [Perez and Dragicevic, 2009]. Two bottom-up approaches have been proposed to deal with this issue.

The first technique, emerged from the Cellular Automata (CA) theory, has been adopted to model location-specific attributes of susceptible populations along with stochastic parameters that capture the probabilistic essence of the disease transmission [Sirakoulis et al., 2000]. CAs [Von Neumann and Burks, 1966] are models of physical systems, where space and time are discrete and interactions are local. They have been broadly used as models for complex systems and applied to several physical problems governed by local interactions [Karafyllidis, 2004; Karafyllidis and Thanailakis, 1997] . A CA comprises a regular uniform n-dimensional lattice or array as illustrated in Figure 2.11. At each site of the lattice (cell), a physical quantity takes on values. This physical quantity is the global state of the CA, and the value of this quantity at each site is its local state. Each cell is confined to local neighborhood interaction only [Wolfram, 1994].

Nevertheless, the CA model is unable to describe the individual's movement and interactions over the transmission space. This is an essential element to consider especially in highly contagious diseases. Agent-Based Modeling (ABM), a solution similar to the CA methods, models a system by representing each individual or agent in that system, and addresses this issue as it has the exceptional capability of tracking the movement of a disease and the contacts between individuals of a social group located in a specific geographic area [Patlolla et al., 2006]. It allows the study of specific spatial characteristics of the disease transmission as

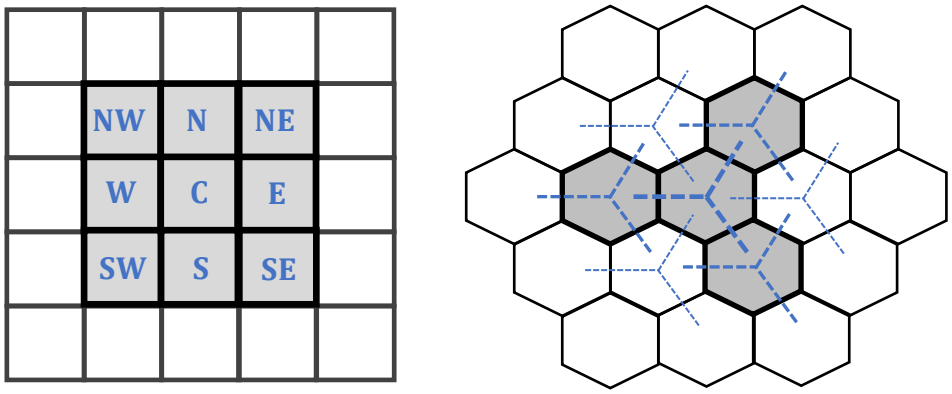


Figure 2.11: Distinct types of neighborhood, where the gray cells represent the size of the contact neighbors.

well as the stochastic nature of the epidemic process. The Agent-Based (AB) concept consists of a population of individual "agents", an environment, and a set of rules (fixed or mutable). In ABM the actions occur through the agents, which are simple and self-contained programs that gather information from their surroundings and use it to decide how to act [Heppenstall et al., 2011]. The actions can be as straightforward as deciding which direction it will move in based on a simulated perception, or they can be more complex, like looking for other agents within a certain radius who share specific characteristics and socially interacting with them. Unexpected aggregate phenomena that emerge from a model's combination of individual behaviors can be captured by the ABM [Bruch and Atwell, 2015b].

For instance, in the proposed approach [Zhen and Quan-Xing, 2006], the natural biological process of a disease spreading among people is characterized as well as the daily behaviors of the individuals in an urban environment. Georeferenced Geographic Information Systems (GIS) data layers of an urban region are used to create the model in order to geographically reflect the typical urban landscape. In addition it was created to contain georeferenced data about population densities, various land uses, and transportation networks in order to take into consideration some of the aspects that may affect an epidemic in metropolitan conditions. Figure 2.11 illustrates the AB process of the mentioned method.

Overall this technique brings various advantages since it tracks the disease progression down to each individual, which allows it to follow individual contacts in social networks and geographical areas such as universities. Also it is possible to introduce heterogeneity and diversity of the target population in contrast to compartmental models [Iranzo and Pérez-González, 2021b]. ABM is able to explore changes in people's behavior as a result of the introduction of a particular intervention [Bruch and Atwell, 2015a].

They also carry some drawbacks in comparison to equation-based approaches. For more complex models, there is the lack of use of real landscape structures and integration with geospatial data and GIS to represent the continuous environment where the discrete individuals interact. Also, complex ABM involve innumerable parameters that must be empirically calibrated, yet it is common

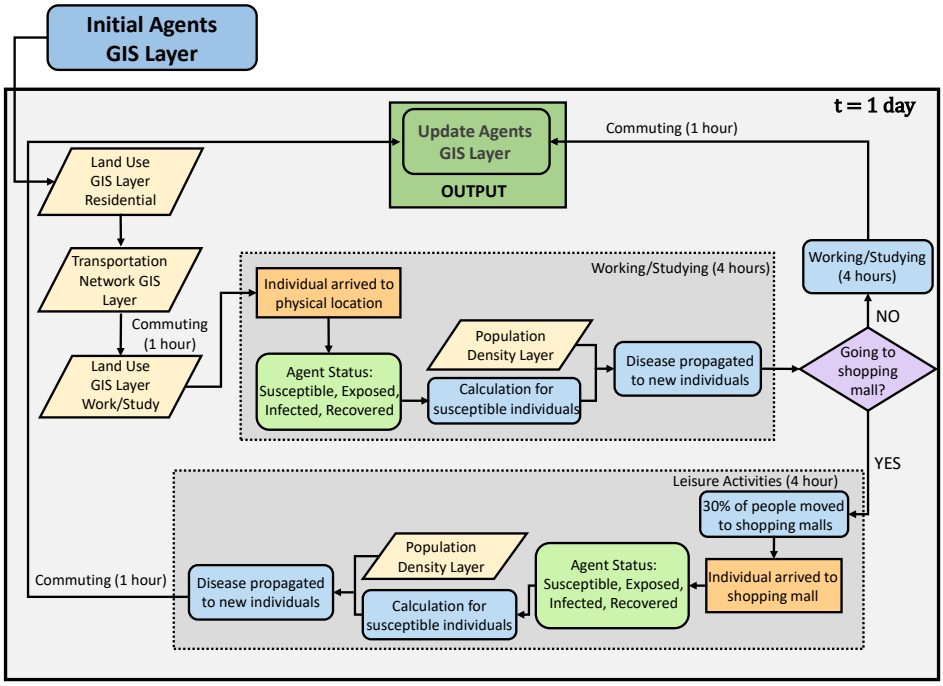


Figure 2.12: Illustration of the AB process for a single time set representing their daily activities and interactions with the environment [Perez and Dragicevic, 2009].

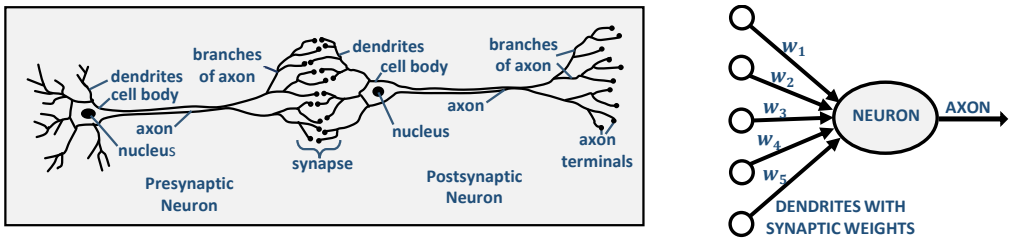
the scenario where the available data about some of those parameters is not reliable. High-fidelity ABM portraits many interactions and dynamic processes which adds a possible uncertainty around the model results, difficulties the models interpretation and validation [Hunter et al., 2020], and results or conclusions drawn from them are hardly to generalize [Iranzo and Pérez-González, 2021a]. Additionally, ABM has a high computational cost.

2.3 Artificial Neural Networks

In this section we present the two artificial neural networks architectures considered to sort out the graph learning problem. Subsection 2.3.1 details the basics of ANNs and drawbacks of simple Feed-Forward Neural Networks (FFNNs), and 2.3.2 introduces the core functioning of CNNs.

2.3.1 On the Basics of Artificial Neural Networks

ANNs are a popular machine learning technique which evolved from the idea of emulating the human brain: (i) neurons receive signals from the dendrites; (ii) the signal is processed and an output signal might be sent through the axons. Similarly, an ANN takes inputs from the input layer, weighs them individually and passes them through a nonlinear activation function in order to produce an output [Aggarwal, 2018]. Fig. 2.13 shows the comparison between a biological neural network and an artificial neural network with a single neuron.



(a) Biological neural network (b) Artificial neural network

Figure 2.13: Relation between biological and artificial neural networks [Aggarwal, 2018].

An ANN is composed of three main components: the node structure, the network topology and the learning rule [Zou et al., 2008]. Here we focus on FFNNs (also designated Multi-layer Perceptrons) where each layer feeds its state into the next layer, from the input towards the output.

The node is the basic processing element in a neural network. The node structure, illustrated in Fig. 2.13(b), dictates how the information is processed. Its structure consists of the inputs x_i connected to the node, the weights w_{ij} that linearly combine the inputs, the activation function f, the bias w_0 and the output O_i. The activation function defines how the weighted sum of the input is transformed into an output and is chosen based on the modeling problem.

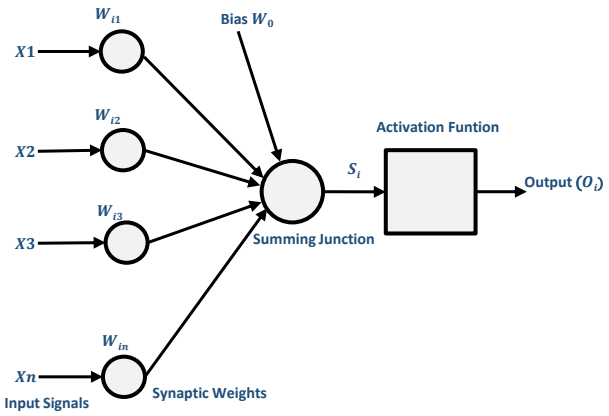


Figure 2.14: Structure of an artificial neural network node.

The output of the i-th neuron N_i is given mathematically by the following expression

$$O_i = f\left[w_0 + \sum_{j=1}^{n} w_{ij}x_j\right]. \tag{2.18}$$

The network topology is associated with the way the nodes are disposed and connected. Figure 2.15 shows the general architecture of a multi-layer FFNN. The nodes are organized into linear arrays, designated layers. The first and last layers are termed input and output layers, whereas the one in the middle are the hidden layers.

The learning rule defines how the weights are initialized and adjusted. Learning is classified into two main categories, supervised and unsupervised learning. The

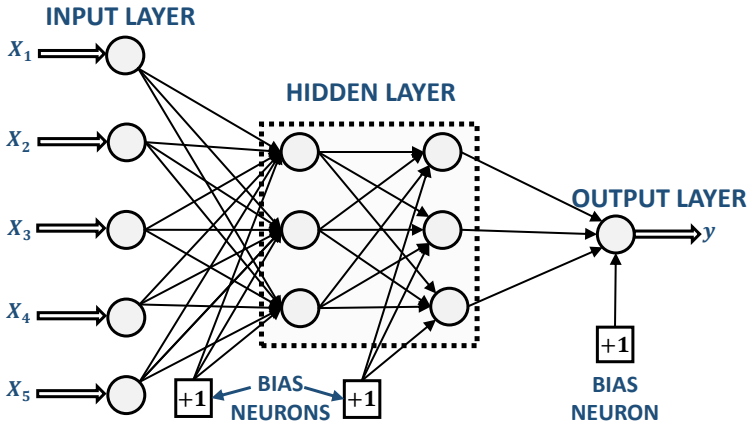


Figure 2.15: Architecture of a feed-forward neural network with two hidden layers and a single output layer.

training of an ANN lies in supervised training, i.e., we need to provide training data that contains examples of inputs and the expected output. The training set must incorporate representative samples of the underlying model, otherwise the trained model will not be reliable. The goal of the training is to minimize the error function by adjusting the weights connecting the neurons.

Regarding the loss function, the idea is to give a high penalty for wrong predictions and a low penalty otherwise. For regression problems the error function is generally given by the Mean Squared Error (MSE) between the neural network output and the real output Y_i for a number of N samples

$$E_{MSE}(W) = \frac{1}{N}\sum_{i=1}^{N}(f_W(X_i) - Y_i)^2, \tag{2.19}$$

where W is the vector of weights of the neural network and Y_i is the output associated with the input X_i.

For classification problems the cross-entropy loss function may be more appropriate. Anyhow, in theory a ANN can be trained equally well by minimizing either function, considering it is capable of approximating the true posterior distribution arbitrarily [Golik et al., 2013].

While the error function defines the metric used to improve the model, the process responsible for adjusting the error and minimizing the error is called *backpropagation*. This algorithm contains two steps: the forward and backward phase. During the forward phase inputs of the sample are fed to the neural network, the predicted value is compared to the real output and the derivative of the error function regarding the output is computed. Amid the backphase, the gradient of the loss regarding the different weights is learned. Those gradients are needed to adjust the weights according to a learning rate α as it follows

$$W_{new} = W_{old} - \alpha\left(\nabla_W E\right) \tag{2.20}$$

Training algorithms are discussed extensively in the literature, including [Rumel-

hart et al., 1986] and [Lin and Lee, 1996] for fundamental concepts.

The accuracy of the neural network training is directly linked with the selection of parameters of the node structure and the network topology. Increasing indiscriminately the number of neurons and/or the number of intermediate layers does not guarantee a proper generalization of the ANNs with respect to the samples of the test subset. Such choices may guide the model to overfit, a condition in which the network memorizes the responses to the input. In this scenario, the error of the training process tends to be very low but when the test subset is presented to the network, the error tends to be very high. Contrastingly, an ANN with a reduced number of neurons might be not sufficient to extract and store the features of the problem. In that case, it will not be able to build hypotheses about the process behavior, which results in underfitting. In this case, the squared error in both training and test stages are very notable [da Silva et al., 2017].

FFNNs bring various advantages to the machine learning field when comparing to other traditional techniques: (i) can be used easily without prior knowledge, (ii) create the required decision function directly through the learning process, (iii) can be used for many fields and tasks as discrimination, pattern recognition, empirical modeling, (iv) can represent both linear and nonlinear relationships [Park and Lek, 2016]. On the other hand, FFNNs require a neuron for each input and the number of weights may quickly become unmanageable for samples of large dimensions. It includes too many parameters because it is fully connected: each node is connected to every other node in the next and the previous layer, resulting in redundancy and inefficiency. Most importantly, they are not suitable for some tasks as spatial information is lost when the data is flattened from a matrix to a vector.

2.3.2 Convolutional Neural Networks

The basic concept behind CNNs is to devise a solution for reducing the number of parameters allowing a network to be deeper with much less parameters, therefore addressing the drawbacks of FFNNs [Aghdam, 2017]. CNNs are a kind of FFNNs that are able to extract features from data with convolution structures. Differently from traditional feature extraction methods, such as Scale Invariant Feature Transform (SIFT) [Lindeberg, 2012] and Local Binary Pattern (LBP) [Ahonen et al., 2006], there is no need to extract features manually.

In comparison with the general artificial neural networks, CNNs have various advantages: (i) each neuron is no longer connected to all neurons of the previous and next layer which is useful to decrease parameters and accelerate convergence; (ii) a group of connections can share the same weights, which reduces parameters further; (iii) the pooling layers, described further, can reduce the amount of data while retaining useful information (data down-sampling), and reduce the number of parameters by removing trivial features [Li et al., 2021].

CNN is a mathematical construct which consists of alternate layers of convolution and pooling followed by one or more fully connected layers at the end. In some cases, a fully connected layer is replaced with a global average pooling layer. The

first two, convolution and pooling layers, perform feature extraction, whereas the fully connected layer, maps the extracted features into final output. The multilayered, hierarchical structure of deep CNNs, allows to extract low, mid, and high-level features. High-level features (more abstract features) are a combination of lower and mid-level features. Figure 2.16 illustrates the Operation of a two-dimensional CNN.

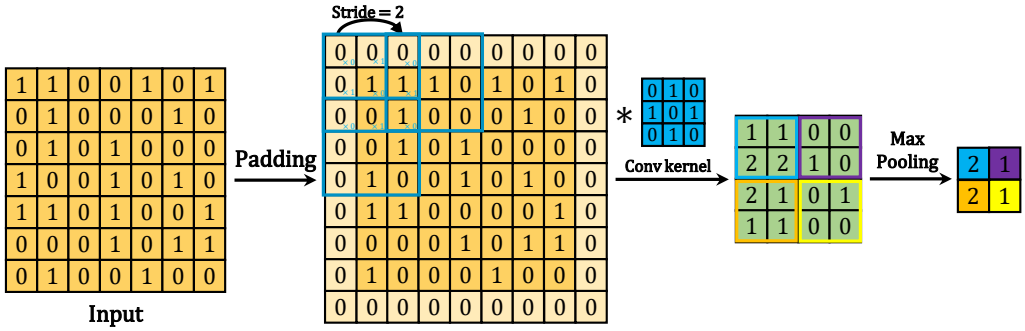


Figure 2.16: Operation of a two-dimensional CNN.

The convolutional layer has a set of convolutional kernels where each neuron behaves like a kernel. The kernel divides the image into small slices, designated receptive fields. It convolves the data using a particular set of weights by multiplying its elements with the corresponding elements of the receptive field [Günther et al., 2014]. Pooling sums up similar information in the neighborhood of the receptive field and outputs the dominant response within the local region [Lee et al., 2016].

2.4 Epidemic Models with Deep Learning

Epidemic models enable the modeling of the spread of infectious diseases, either on a macroscopic or microscopic level. It is also well-known that forecasting the spatial and temporal evolution of an epidemic is currently an active area of research. Particularly, with the emergence of ANNs which contributed to the advance of time-series prediction and forecasting. Here we review some works that combine machine learning techniques and epidemiological causal models. This hybrid model may be applied to either improve the model parameterization or to enhance the forecasting of the pandemic.

Methodologies purely based on deep learning do not allow to make use of the knowledge available in the domain of predictive epidemiology. In addition, these methods are less suitable when it comes to the interpretability of results, especially if those predictions support a decision process. Interpretation and explainability of predictions are critical, e.g., in health related activities [McKelvey et al., 2018].

Reference [Rahmadani and Lee, 2020a] proposed a hybrid deep learning based epidemic prediction framework for the spread of Covid-19. Their approach uses an expanded Susceptible-Exposed-Infected-Susceptible (SEIR) model in order to

capture the disease transmission among distinct populations. Their framework incorporates deep learning with a meta-population model to obtain a more accurate parametric estimation as illustrated in Fig. 2.17.

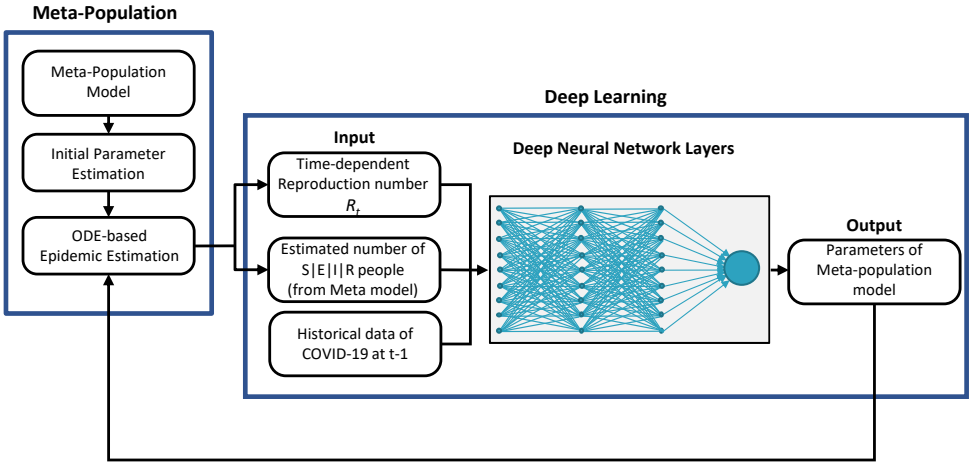


Figure 2.17: Architecture of a feed-forward network with two hidden layers and a single output layer [Rahmadani and Lee, 2020a].

[Wang et al., 2020] proposed a framework that attempts to solve the problem related to the lack of data for diseases such as Influenza-Like diseases, with a data augmentation framework. The data augmentation is performed by generating synthetic data via ODE-based models such as the SEIR model and then using deep neural networks to capture the dynamics. The main idea is to tune the parameters in order to minimize the difference between synthetic and real observed data.

Other related works that make use of deep learning to find the best parameters that minimize the gap between real and synthetic data are: (i) [Jo et al., 2020] proposed a forward-inverse neural network for SIR models where the parameters at each time step are estimated with a time dependent ANN based on the historical data of COVID-19 in South Korea; (ii) [Farooq and Bazaz, 2020] make use of a ANN to learn the compartmental parameters through an incremental learning approach, i.e., the ANNs are used to iteratively approximate the incoming data which allows to improve the model without having to train it every time the dataset is updated.

2.5 Causal Inference of Networked Dynamical Systems

In order to illustrate in a transparent manner the problem of causal inference of networked dynamical systems, we focus on discrete-time linear stochastic networked dynamical systems given by

$$y_i(n+1) = \sum_{j=1}^{N} A_{ij} y_j(n) + x_i(n+1) \tag{2.21}$$

where $y_i(n)$ represents the state of the node i at time n; $\{x_i(n)\}_{i,n}$ is zero-mean and i.i.d spatially (i.e., in the index i) and temporally (i.e., in the index n); A is a nonnegative matrix with spectral radius $\rho(A) < 1$ modeling the interaction among the N nodes. In particular, $A_{ij} \neq 0$ means that the state $y_j(n)$ of node j bears a direct impact on the state $y_i(n+1)$ of the node i at time $n+1$. In other words, the *support* of the interaction matrix A conveys the underlying graph linking the nodes.

One important question is: can we infer A from the observed time-series $\{\mathbf{y}(n)\}_{n=1}^{\infty}$? Indeed, we have that

$$\mathbf{y}(n+1)\mathbf{y}(n)^\top = A\mathbf{y}(n)\mathbf{y}(n)^\top + \mathbf{x}(n+1)\mathbf{y}(n)^\top \tag{2.22}$$

which yields,

$$\mathbb{E}\left[\mathbf{y}(n+1)\mathbf{y}(n)^\top\right] = A\mathbb{E}\left[\mathbf{y}(n)\mathbf{y}(n)^\top\right], \tag{2.23}$$

and thus,

$$\begin{aligned} A &= \mathbb{E}\left[\mathbf{y}(n+1)\mathbf{y}(n)^\top\right]\left(\mathbb{E}\left[\mathbf{y}(n)\mathbf{y}(n)^\top\right]\right)^{-1} \\ &= R_1(n)R_0^{-1}(n) \overset{n\to\infty}{\longrightarrow} R_1(R_0)^{-1}, \end{aligned} \tag{2.24}$$

where $R_1 = \lim_{n\to\infty} 1/n \sum_{i=1}^{n} \mathbf{y}(n+1)\mathbf{y}(n)^\top$ is the limiting 1-lag correlation matrix and $R_0 = \lim_{n\to\infty} 1/n \sum_{i=0}^{\infty} \mathbf{y}(n)\mathbf{y}(n)^\top$ is the limiting correlation matrix. The matrix-estimator $R_1R_0^{-1}$ is often referred to as *Granger* estimator. This provides a transparent scheme to recover the causal relationships from the time-series.

For the discrete-time linear stochastic networked dynamical system (2.21), we are able to derive in closed form an estimator for the underlying interaction matrix A. Such a scheme is not unique. For instance, when the interaction matrix A is symmetric, and the noise is Gaussian with covariance matrix $\sigma^2 I_N$, where I_N is the $N \times N$ identity matrix, then we have that

$$R_0(n) = \sigma^2 \sum_{i=0}^{\infty} A^{2i}, \tag{2.25}$$

and observing that

$$\begin{aligned} R_1(n) = AR_0(n) &= \sigma^2 \textstyle\sum_{i=0}^{\infty} A^{2i+1} \\ R_3(n) = A^3R_0(n) &= \sigma^2 \textstyle\sum_{i=0}^{\infty} A^{2i+3} \end{aligned}, \tag{2.26}$$

which yields the relation

$$R_1(n) - R_3(n) = \sigma^2 A. \tag{2.27}$$

In other words, the support of A can be inferred via subtracting the 3-lag covariance matrix from the 1-lag moment counterpart. This was proposed and explored in [Chen et al., 2022] and it turns out to be a relevant result for Theorem 1 in Chapter 3 yielding the separability of the features.

2.6 Causal Inference of Large-scale Networked Dynamical Systems

For large-scale networks, only some nodes in the network can be observed. Define $\mathcal{S} \stackrel{\Delta}{=} \{1, 2, \ldots, S\}$, with $S < N$, as the set of observable nodes. In other words, in the large-scale setting, we can only observe $\{[\mathbf{y}(n)]_{\mathcal{S}}\}_{n=1}^{\infty}$, where

$$[\mathbf{y}(n)]_{\mathcal{S}} = (y_1(n), y_2(n), \ldots, y_S(n)) \tag{2.28}$$

collects only the entries of the vector $\mathbf{y}(n)$ at the set of observable nodes $\mathcal{S}$. If we insist in adopting the Granger estimator under this partially observed setting, we have the following

$$\widehat{A}_{\mathcal{S}} = [R_1]_{\mathcal{S}} \left([R_0]_{\mathcal{S}}\right)^{-1} \neq \left[R_1 \left(R_0\right)^{-1}\right]_{\mathcal{S}} = A_{\mathcal{S}}, \tag{2.29}$$

where $A_{\mathcal{S}}$ represents the true interaction matrix among the nodes in the observed set $\mathcal{S}$; the latter identity follows from the derivation in the previous subsection; and $\widehat{A}_{\mathcal{S}}$ is the estimated interaction matrix obtained via Granger and ignoring the latent part. Even though $[R_1]_{\mathcal{S}} \left([R_0]_{\mathcal{S}}\right)^{-1} \neq \left[R_1 \left(R_0\right)^{-1}\right]_{\mathcal{S}}$, i.e., part of the information about the true interaction matrix $A_{\mathcal{S}}$ is lost in the partial observability regime by the latent part of the network, under certain regimes of network connectivity, the underlying network structure is preserved in $\widehat{A}_{\mathcal{S}}$ in that under an appropriate thresholding of the entries, we can recover the underlying network [Santos et al., 2020a].

In this thesis, we explore this causal inference under partial observability via ANNs, as well: can we train neural networks in order to recover the underlying network structure under partial observability from the observed time-series? As we will show, the answer is yes, and we provide an efficient mechanism to extract this structural information.

2.7 Ill-posed Nature of Network Inference

Inferring the interactions and causal relationships among nodes governed by a networked dynamical system from the observed time-series data is a demanding task. Various mathematical models have been proposed to address this problem, however causal inference is far from being a closed issue, particularly for nonlinear dynamic systems [Stepaniants et al., 2020]. In particular, it is typically a high-dimensional inverse problem that is ill-posed, in general.

Inferring causal relations using the time series data alone may lead to distinct solutions that accurately reproduce the data. Namely, identical time series can stem from distinct network topologies, rendering the structure inference an ill-posed problem, in general, as illustrated in Fig. 2.18.

The solution to this issue generally lies on perturbing the networked dynamical system and as a result, observing the transient dynamics of relaxation back to

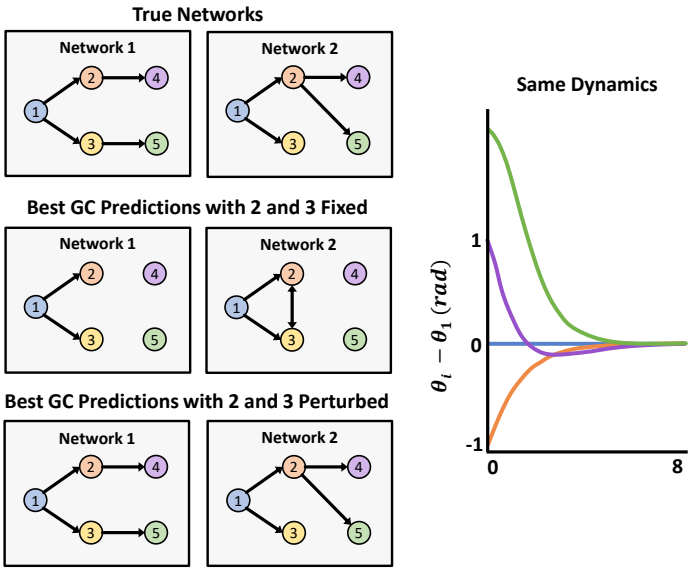


Figure 2.18: Illustration of the ill-posed nature of network inference. [Stepaniants et al., 2020].

the original equilibrium, i.e., measure its collective response, in order to infer the underlying structure, as illustrated in Fig. 2.19.

For instance, [Stepaniants et al., 2020] proposed two methods for nonlinear dynamical systems using the idea of perturbing the system: judiciously perturb Granger Causality (GC) and call Perturbation Cascade Inference (PCI). The first approach uses the base GC model, however the system is perturbed giving all nodes of the network random initial conditions and the time series is sampled from the transient dynamics measurements. PCI is an active inference approach that learns the distance from every node to the initially perturbed one and applies that knowledge to rebuild the connectivity of the underlying network.

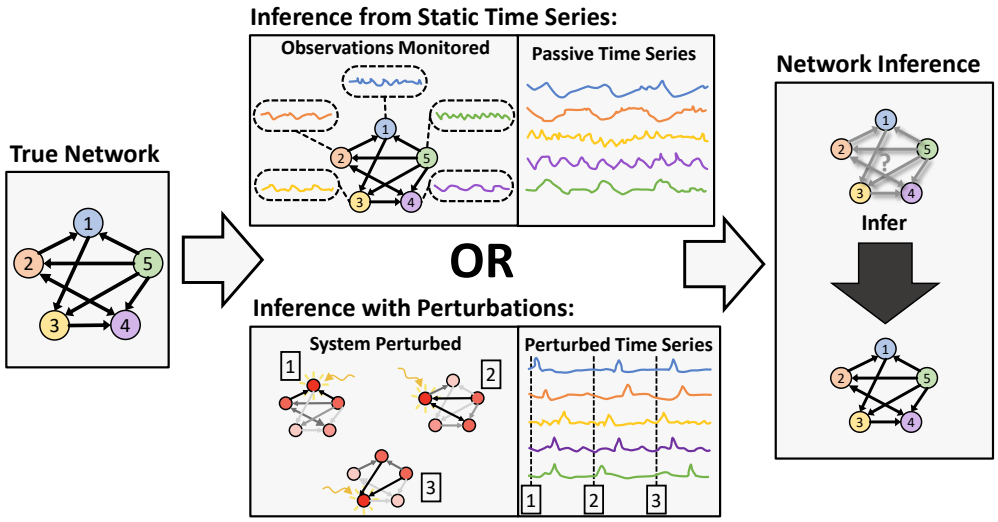


Figure 2.19: Illustration of two approaches concerning network inference [Stepaniants et al., 2020].

Chapter 3

Related Work

This section respects the structure of the related work reported in the submitted manuscript [Machado et al., 2022].

The generative process underlying the nature of the time series samples determines quite critically the method used to infer the graph of interactions from the observed time series data. For example, if the observed time series samples follow a Gaussian multivariate distribution and are independent and identically distributed (i.i.d.), then the Precision matrix (inverse of the covariance matrix) is a consistent estimator for the network structure. On the other hand, if the time series reflect the state evolution of a linear stochastic networked dynamical system (not i.i.d.), then regression (or Granger) is consistent [Geiger et al., 2015; Matta et al., 2020] and Precision matrix no longer grants structural consistency. If further, the time series samples are i.i.d. following a ferromagnetic Ising model distribution, then the correlation matrix is a possible structurally consistent matrix estimator [Montanari and Pereira, 2009], even though, in general, it is a poor estimator for other networked systems.

Not only the underlying law governing the samples is relevant, but also the observability of the system. Whether the system is fully or partially observed conforms to important information to assert the causal inference method (if any) to consistently and optimally extract structural information of the networked system. Next, we stratify some relevant related works accordingly. Section 3.1 addresses the methods present in the literature capable of recovering the graph of interactions under full observability. Section 3.2 reviews the methods capable of recovering the structural connectivity only when a subset of time series data is available (partial-observability).

3.1 Full-observability

Graphical Models. These models focus on samples that are i.i.d. and stem from multivariate distributions. Some relevant and well-known algorithms include SGS [Spirtes et al., 2000], PC [Spirtes and Glymour, 1991], GES [Chickering, 2003], and FGS [Ramsey et al., 2016]. These methods are fit to networks of *reasonable*

size. Reference [Anandkumar et al., 2012] offers a method for the large-scale network setting via conditional covariance tests. The commonality among all these works is the demanding structural constraints (revolving around sparsity of connections) to grant the consistency of the methods. In other words, the consistency of the methods is only granted if the underlying network (object of inference) is sparse enough. The control over the underlying connectivity of the ambient network in the referred methods is critical, otherwise the inference problem quickly becomes hard or unfeasible [Bogdanov et al., 2008; Bresler et al., 2014].

Networked dynamical systems. Reference [Mateos et al., 2019] offers a survey of recent results for full-observability over different models (in particular, linear models), focused on enforcing sparsity of the underlying network, which includes linear VAR models such as [Mei and Moura, 2017]. In this line of work, reference [Pereira et al., 2010] addresses linear stochastic differential equations (SDEs) via an optimization formulation that primarily enforces sparsity of the network. Therefore, the performance of the method strongly depends on the sparsity of the network. The problem is, in fact, addressed over a discrete-time model (matching the same model adopted in this thesis) obtained from the continuous-time SDE. Other strategies, such as [Granger, 1969; Segarra et al., 2017a,b], leverage on spectral properties of the underlying interaction matrix, namely, via finding signatures in the time series that are closely related with the spectrum of the interaction matrix in the corresponding generative process. These methods also rely on the sparsity of the network and are quite sensitive to the observability of the system: as soon as a subset of the networked system is not observable, the system provides no guarantees of consistency at all.

3.2 Partial-observability

The framework for graph structure identification in the context of partial observability (i.e., in the presence of latent variables) is more challenging than the full-observability counterpart and deserves separate attention. It is important to highlight that the majority of the works under partial observability belong to the literature of graphical models – samples are assumed i.i.d and drawn from a joint probability distribution.

Graphical Models. The techniques frequently rely on Conditional Independence (CI) tests. Well-known algorithms for causal inference under the presence of latent variables are the FCI [Spirtes et al., 1995] and RFCI [Colombo et al., 2012]. As in the full-observability scenario, the performance of the methods scales poorly with the connectivity of the underlying interaction graphs rendering these CI-based approaches impractical for dense graphs. The methods tend to impose several structural constraints to design sufficient conditions for structural consistency as directed acyclic property (no loops), long girth [Anandkumar and Valluvan, 2013; Anandkumar et al., 2013] and other more technical structural conditions as bottleneck and non-redundancy [Adams et al., 2021; Mastakouri et al., 2021]. These properties imply that the network should be quite sparse.

Networked dynamical systems. References [Materassi and Salapaka, 2012a,b,

2015] address linear networked dynamical systems using certain pseudo-metrics (such as log-coherence distance) computed from the time series samples that are intended to estimate the distance between nodes in the underlying graph. As long as the network complies with a number of rigorous sparsity requirements, such as not allowing undirected cycles (which excludes, e.g., graphs obtained from the realization of Erdős–Rényi random graph models), it is demonstrated that some pairs of nodes may be consistently categorized. Reference [Geiger et al., 2015] offers constraints on the network connectivity and interaction matrix of a linear networked dynamical system to achieve uniqueness of the network connectivity given partially observed time series samples. While it offers a *uniqueness* result, that is, that the interaction matrix is uniquely determined by the partially observed time series data, it does not offer an algorithm with consistency guarantees to infer the network. Reference [Zhao and Wan, 2022] addresses specific discrete-time discrete state-space networked dynamical systems using an Expectation-Maximization based methodology. References [Chandrasekaran et al., 2012; Jalali and Sanghavi, 2012; Mei and Moura, 2018] uses convex optimization-based techniques to regularize the network's sparsity under partial observability. References [Matta et al., 2020, 2022; Santos et al., 2020b] prove structural consistency of the Granger (or regression) estimator over various regimes of network connectivity, including dense networks, over partially observed discrete-time linear stochastic networked dynamical systems. Similar to [Anandkumar and Valluvan, 2013], these estimators are structurally consistent in the *thermodynamic limit*, that is, when the number of nodes scales to infinity, which fits the framework of large-scale networks. In a recent work, [Chen et al., 2022] proved that the underlying interaction matrix can be expressed as a linear combination of covariance matrices (if enough time series samples are available), under the following regime: (i) the interaction matrix A is symmetric; (ii) the excitation noise is *diagonal* and homogeneous, that is, its covariance matrix is a multiple of the identity matrix. Theorem 1 in [Chen et al., 2022] will be useful to establish an important result regarding the separability of the features proposed.

Chapter 4

Formal Analysis and Methodology

In this chapter, we lay down the formal analysis for our proposed graph learning method. In particular, we propose a set of features as statistical descriptors for the connectivity of the pairs of nodes, i.e., from the time series samples stemming from each pair, we compute a feature vector for the corresponding pair. We prove that the set of proposed features is linearly separable, that is, there exists a hyperplane that separates the feature vectors associated with connected pairs from the feature vectors associated with disconnected pairs. This motivates the methodology of using these features to train CNNs in order to assert the connectivity of a pair.

The lemmas and theorems, described here, have been reported in the submitted manuscript [Machado et al., 2022] and the notation and structure of this section follow that of the reported manuscript. Further, we include the proofs to the lemmas and theorems in order to render the thesis self-contained.

Section 4.1 formulates the problem and introduces the notation for the rest of the chapter. Section 4.2 provides some useful definitions of structural consistency of matrix-valued estimators and feature-based estimators (sometimes referred to as tensor-valued estimators). Section 4.3 proves the linear separability of the features. Finally, Section 4.4 describes the methodology of our approach to be further explored in the numerical experiments.

4.1 Problem Formulation

The focus of this thesis will be on linear stochastic networked dynamical systems. Notwithstanding, the tools developed in the linear framework can be also useful over a great class of nonlinear networked dynamical systems. Indeed, several nonlinear dynamical systems exhibit an approximate linear dynamics when close to the equilibrium (for those converging to an equilibrium) and graph learning over this family of nonlinear systems are often dealt with via linearization about the equilibria under small-noise regimes [Ching and Tam, 2017], or via an appropriate embedding in higher-dimensional spaces [Mauroy and Goncalves, 2016] in order to yield a linear system. It is also common to consider discrete-

time dynamics as result of a time discretization process as in [Montanari and Pereira, 2009], where the time-discretization of solutions to continuous time linear stochastic differential equations yield the discrete-time linear model (refer to equation (4.1)) adopted in this thesis. Moreover, graph learning over linear networked dynamical systems is still an active area of research with several open questions.

Next, we consider the linear networked dynamics given by

$$\mathbf{y}(n+1) = A\mathbf{y}(n) + \mathbf{x}(n+1), \tag{4.1}$$

where $\mathbf{y}(n) = [y_1(n)\ y_2(n)\ \ldots\ y_N(n)]^\top \in \mathbb{R}^N$ is a vector that represents the state of the N-dimensional networked dynamical system at time n, i.e., it collects the states $y_i(n)$ of each node i at time n; $\mathbf{x}(n) \sim \mathcal{N}\left(0, \sigma^2 I_N\right)$ is the excitation noise or perturbation associated with the N nodes of the system, with *diagonal* covariance matrix $\sigma^2 I_N$, where I_N is the identity matrix, and it is assumed independent across time n; $A \in \mathbb{R}_+^{N\times N}$ refers to the non-negative interaction matrix whose support represents the underlying graph linking the nodes. We assume that the dynamical system (4.1) is stable, which translates into assuming $\rho(A) < 1$, where $\rho(A)$ stands for the spectral radius of A, i.e., the greatest of the absolute value of its eigenvalues.

We address the problem of consistently recovering the support of the submatrix A_S, or equivalently, the underlying graph structure of interactions among the observed nodes in the subset S from the time series represented by the observed subvector $[\mathbf{y}(n)]_S = \left[\mathbf{y}_{m_1}(n)\ \mathbf{y}_{m_2}(n)\ \ldots\ \mathbf{y}_{m_{|S|}}(n)\right]^\top \in \mathbb{R}^{|S|}$ over time n, where $|S|$ is the cardinality of the subset S, as illustrated in Fig. 1.1, in Chapter 1.

Notation: $S = \left\{m_1, m_2, \ldots, m_{|S|}\right\} \subset \{1, 2, \ldots, N\}$ is a nonempty subset of indexes with $m_1 < m_2 < \ldots < m_{|S|}$ and $|S| \leq N$ and will represent the subset of observed nodes; given a vector $\mathbf{y} \in \mathbb{R}^N$, $[\mathbf{y}]_S = \left[\mathbf{y}_{m_1}(n)\ \mathbf{y}_{m_2}(n)\ \ldots\ \mathbf{y}_{m_{|S|}}(n)\right]^\top$ is the subvector obtained from $\mathbf{y}$ and indexed by S; we adopt a similar notation for matrices, namely, given $A \in \mathbb{R}^{N\times N}$, the matrix $A_S \in \mathbb{R}^{|S|\times|S|}$ or $[A]_S \in \mathbb{R}^{|S|\times|S|}$ is defined as the submatrix whose ij^{th} entry is $A_{m_i m_j}$; $\text{Supp}\,(A)$ is the support of the matrix A, i.e., $[\text{Supp}\,(A)]_{ij} = \mathbf{1}_{\left\{A_{ij}\neq 0\right\}}$; $||\mathbf{y}||_\infty$ refers to the L_∞-norm that returns the maximal absolute value across the entries of the vector $\mathbf{y} \in \mathbb{R}^N$; the set of natural numbers is denoted by $\mathbb{N} = \{0, 1, 2, \ldots\}$.

4.2 Structural Consistency

Next, we introduce important building blocks to the problem of graph learning, namely, the so-called k-lag covariance matrices that are defined as

$$R_k(n) \stackrel{\Delta}{=} \mathbb{E}\left[\mathbf{y}(n+k)\mathbf{y}(n)^\top\right] \tag{4.2}$$

and are associated with the process $(\mathbf{y}(n))_{n\in\mathbb{N}}$. Further, we define their empirical counterparts given by

$$\widehat{R}_k(n) \triangleq \frac{1}{n}\sum_{\ell=0}^{n-1}\mathbf{y}(\ell+k)\mathbf{y}(\ell)^\top. \tag{4.3}$$

We will also refer to these latter k-lag empirical covariance matrices simply as *k-lag moments*.

These covariance matrices entail relevant structural information and when properly combined can yield a useful methodology for graph learning. In particular, our feature vectors will be built upon them.

We refer to a matrix-valued estimator as any map whose input is given by the (observed) time series and the output is given by a matrix, namely,

$$\begin{array}{rccc} F^{(n)}: & \mathbb{R}^{|S|\times n} & \longrightarrow & \mathbb{R}^{|S|\times|S|} \\ & \{[\mathbf{y}(\ell)]_S\}_{\ell=0}^{n-1} & \longmapsto & \mathcal{F}^{(n)} \end{array}, \tag{4.4}$$

for any given $n\in\mathbb{N}$. The core idea is that the ij^{th} entry of the output matrix $\mathcal{F}^{(n)}$ estimates the strength of the link from i to j from n time series samples. For instance, the k-lag moment matrix $\widehat{R}_k(n)\in\mathbb{R}^{N\times N}$ just introduced in equation (4.3) or the $\left[\widehat{R}_k(n)\right]_S \in \mathbb{R}^{|S|\times|S|}$ submatrix, in the case of partial-observability, are examples of matrix-valued estimators as they are matrices computed from the time series as in equation (4.3).

Given a sequence of random variables $Z^{(n)}$, we say that $Z^{(n)} > \tau$ with high probability (w.h.p.), whenever

$$\mathbb{P}\left(Z^{(n)} > \tau\right) \overset{n\to\infty}{\longrightarrow} 1. \tag{4.5}$$

Roughly speaking, if n is large enough, then the probability that $Z^{(n)}$ lies above τ is close to 1. For the sake of simplicity, we might simply refer to this as "$Z^{(n)} > \tau$ with high probability".

Definition 4.1 (structural consistency of a matrix-valued estimator)**.** *A matrix-valued estimator $F^{(n)}$ is structurally consistent with high probability (w.h.p.), whenever there exists a threshold τ so that,*

$$\mathbb{P}\left(\mathcal{F}_{ij}^{(n)} > \tau\right) \overset{n\to\infty}{\longrightarrow} 1 \iff i \to j, \tag{4.6}$$

i.e., i links to j if and only if the ij^{th} entry of the estimator matrix $\mathcal{F}^{(n)}$ lies above the threshold τ, provided that there is a large enough number of samples n.

In other words, up to a proper threshold τ, and computed with enough time series samples n, the output matrix $\mathcal{F}^{(n)}$ contains full-information about the graph of interactions in that $[\mathsf{Supp}(A_S)]_{ij} = \mathbf{1}_{\left\{\mathcal{F}_{ij}^{(n)}>\tau\right\}}$, for all pairs $i\neq j$ w.h.p., as illustrated in Fig. 4.1.

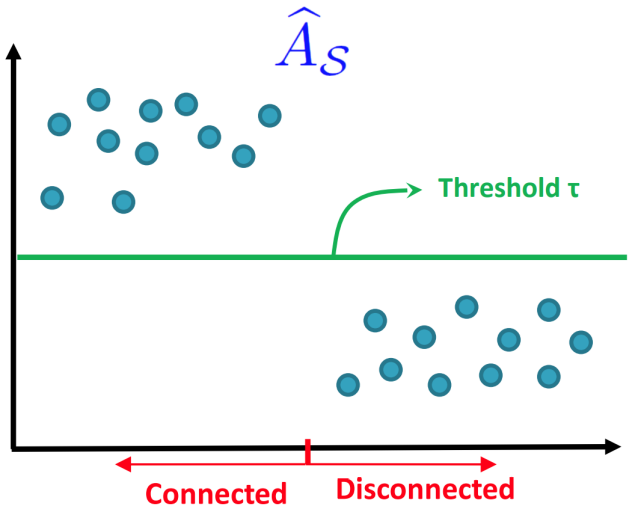


Figure 4.1: Illustration of the structural consistency of a matrix-valued estimator $\widehat{A}_S$. We depict the entries of the matrix-valued estimator $\widehat{A}_S$. Each point in the abscissa indexes a pair in the network and the indexation is so that connected pairs lie on the left-hand side (of the red mark) and disconnected pairs lie on the right. A matrix-valued estimator is structurally consistent when any entry associated with connected pairs lies above any entry associated with disconnected pairs. Equivalently, there exists a threshold that consistently separates the entries.

An example of a structurally consistent w.h.p. matrix-valued estimator (under partial observability) is given by $\mathcal{F}^{(n)} \triangleq \widehat{R}_1(n) - \widehat{R}_3(n)$ [Chen et al., 2022]. Other examples of matrix-valued estimators that are provably structurally consistent under partial observability include:

- **Granger** $\left[\widehat{R}_1(n)\right]_S \left(\left[\widehat{R}_0(n)\right]_S\right)^{-1}$
- **One-lag** $\left[\widehat{R}_1(n)\right]_S$
- **Residual** $\left[\widehat{R}_1(n)\right]_S - \left[\widehat{R}_0(n)\right]_S$

These three matrix-valued estimators are proven to be structurally consistent in the limit of large networks [Matta et al., 2022], i.e., structural consistency is met in the limit $N \longrightarrow \infty$.

We should formally refer to the sequence $\left(F^{(n)}\right)_{n\in\mathbb{N}}$ of maps as being structurally consistent with high probability, as the term "with high probability" introduced before is defined for a sequence of random variables. However, for simplicity of notation, we will simply refer to it as "the estimator $F^{(n)}$ is structurally consistent w.h.p.".

Next, we introduce a tensor-valued estimator which is, formally, any map whose input is given by the (observed) time series and the output is an order-3 tensor, as follows

$$\begin{array}{rccc} T^{(n)}: & \mathbb{R}^{|S|\times n} & \longrightarrow & \mathbb{R}^{|S|\times|S|\times K} \\ & \{[\mathbf{y}(\ell)]_S\}_{n=0}^{n-1} & \longmapsto & \mathcal{T}^{(n)} \end{array}, \tag{4.7}$$

where the ij^{th} entry of the order-3 tensor $\mathcal{T}^{(n)}$ is a vector $\mathcal{T}_{ij}^{(n)} \in \mathbb{R}^K$ that models a feature statistical descriptor assigned to the pair ij in the network and that is built from n time series samples $\{[\mathbf{y}(\ell)]_S\}_{\ell=0}^{n-1}$.

Alike the separability of the entries of a matrix-valued estimator yielding its structural consistency (i.e., the graph of interactions is entailed in the support of the matrix up to a thresholding), we introduce the concept of (linear) structural consistency of a tensor-valued estimator which is equivalent to having the underlying features describing the tensor as linearly separable.

Definition 4.2 (structural consistency of a tensor)**.** *A tensor-valued estimator $T^{(n)}$ of order-3 is linearly structurally consistent with high probability, if there exists an affine map $\mathcal{L} : \mathbb{R}^K \to \mathbb{R}$ (or hyperplane) that separates the underlying features associated with connected pairs from those associated with disconnected pairs w.h.p., that is,*

$$\begin{array}{ll} \mathbb{P}\left(\mathcal{L}(\mathcal{T}_{ij}^{(n)}) > 0\right) \overset{n\to\infty}{\longrightarrow} 1, & \text{if } ij \text{ is connected,} \\ \mathbb{P}\left(\mathcal{L}(\mathcal{T}_{ij}^{(n)}) \leq 0\right) \overset{n\to\infty}{\longrightarrow} 1, & \text{if } ij \text{ is disconnected} \end{array} . \tag{4.8}$$

Equivalently, one can say that the set of features $\left\{\mathcal{T}_{ij}^{(n)}\right\}_{ij}$ is *consistently linearly separable* w.h.p. This means in particular, that if we have access to the separating hyperplane, then we can classify consistently the connectivity of the pairs in the network. As an example, the estimator $T^{(n)}$ whose ij^{th} entry of the tensor output $\mathcal{T}^{(n)}$ is defined as

$$\mathcal{T}_{ij}^{(n)} \triangleq \left(\left[\widehat{R}_D(n)\right]_{ij}, \left[\widehat{R}_{D+1}(n)\right]_{ij}, \ldots, \left[\widehat{R}_L(n)\right]_{ij}\right)$$

corresponds to an order-3 tensor-valued estimator. As we will show in the next section (and it was proved in [Machado et al., 2022]), if $D \leq 1$ and $L \geq 3$, then it is linearly structurally consistent w.h.p., i.e., the set of underlying features $\left\{\mathcal{T}_{ij}^{(n)}\right\}$ is consistently linearly separable w.h.p.

Motivated by this separability property of an order-3 tensor-valued estimator, a normalized version of these features will be used to train CNNs and successfully recover the connectivity pattern of synthetically generated and real-world networks. The proposed framework is summarized in Fig. 4.2

4.3 Features Separability

The technical results presented in this section, regarding the features separability, are the key results supporting the proposed CNN-based framework for graph structure identification of linear networked dynamical systems.

Assumption 1. *Let $\mathcal{E}^{(n)} := \left\{E_1^{(n)}, E_2^{(n)}, \ldots, E_M^{(n)}\right\}$ be a family of matrix-valued estimators such that for some $\mathbf{w} := (w_1, w_2, \ldots, w_M) \in \mathbb{R}^M$ with $\mathbf{w} \neq 0$, the linear*

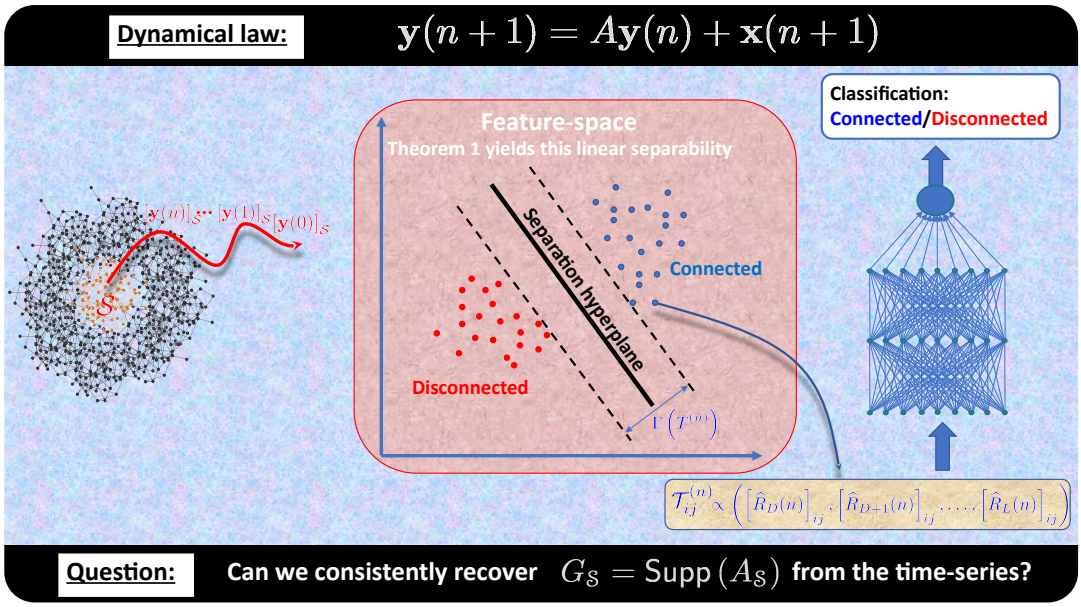


Figure 4.2: Proposed framework: each feature is computed from the time series of each pair of nodes in the linear networked dynamical system. Provided that we have an enough number of time series samples, the set of features produced is linearly separable, i.e., there exists a hyperplane to partition this set consistently into features associated with connected pairs and features associated with disconnected pairs. These features are used to train CNNs to perform classification.

combination $E^{(n)}(\mathbf{w}) = \sum_{\ell=1}^{M} w_\ell E_\ell^{(n)}$ *is a structurally consistent w.h.p. matrix-valued estimator for the dynamics* (4.1).

Lemma 1. *For each pair* ij*, with* $i \neq j$*, define the associated feature vector as,*

$$\mathcal{T}_{ij}^{(n)} := \left(\left[E_1^{(n)}\right]_{ij}, \left[E_2^{(n)}\right]_{ij}, \ldots, \left[E_M^{(n)}\right]_{ij} \right) \in \mathbb{R}^M. \tag{4.9}$$

Then, under Assumption 1, the tensor-valued estimator $T^{(n)}$ *is linearly structurally consistent w.h.p., or equivalently, the set of features* $\left\{\mathcal{T}_{ij}^{(n)}\right\}_{i\neq j} \subset \mathbb{R}^M$ *is consistently linearly separable w.h.p.*

Proof. Since $E^{(n)}(\mathbf{w}) = \sum_{\ell=1}^{M} w_\ell E_\ell^{(n)}$ is structurally consistent w.h.p. for some $\mathbf{w} \in \mathbb{R}^M$, then there exists a threshold $\tau_{\mathbf{w}}$ so that $\left[E^{(n)}(\mathbf{w})\right]_{ij} > \tau_{\mathbf{w}}$ across connected pairs ij and $\left[E^{(n)}(\mathbf{w})\right]_{ij} < \tau_{\mathbf{w}}$, otherwise. Therefore, the affine map $\mathcal{L}_{\mathbf{w}}(\mathbf{x}) = \mathbf{x}\cdot\mathbf{w} - \tau_{\mathbf{w}}$ consistently separates the set of features $\left\{\mathcal{T}_{ij}^{(n)}\right\}_{ij}$ w.h.p. Indeed,

$$\mathcal{L}_{\mathbf{w}}(\mathcal{T}_{ij}^{(n)}) = \mathcal{T}_{ij}^{(n)} \cdot \mathbf{w} - \tau_{\mathbf{w}} = [E(\mathbf{w})]_{ij} - \tau_{\mathbf{w}} > 0 \tag{4.10}$$

for a connected pair ij or

$$\mathcal{L}_{\mathbf{w}}(\mathcal{T}_{ij}^{(n)}) = \left[E^{(n)}(\mathbf{w})\right]_{ij} - \tau_{\mathbf{w}} < 0, \tag{4.11}$$

otherwise. In other words, the hyperplane characterized by the linear map $\mathcal{L}_{\mathbf{w}} : \mathbb{R}^M \longrightarrow \mathbb{R}$ separates consistently the pairs ij for all $i \neq j$. □

Theorem 1. *For each pair ij, with $i \neq j$, define the associated feature vector as,*

$$\mathcal{T}_{ij}^{(n)} := \left(\left[\widehat{R}_D(n) \right]_{ij}, \left[\widehat{R}_{D+1}(n) \right]_{ij}, \ldots, \left[\widehat{R}_L(n) \right]_{ij} \right),$$

with $D \leq 1$ and $L \geq 3$, and assume that the interaction matrix A underlying the dynamics (4.1) *is symmetric and the covariance matrix of the noise process $(\mathbf{x}(n))_{n\in\mathbb{N}}$ is given by $\Sigma_x := \sigma^2 I_N$, for some $\sigma > 0$. Then, the set $\left\{ \mathcal{T}_{ij}^{(n)} \right\}_{i\neq j} \subset \mathbb{R}^M$ is consistently linearly separable w.h.p.*

Proof. Define the vector $\mathbf{w} \in \{-1, 0, 1\}^M$ so that $E^{(n)}(\mathbf{w}) = \widehat{R}_1(n) - \widehat{R}_3(n)$, which is possible since $D \leq 1$ and $L \geq 3$. According to Theorem 1 in [Chen et al., 2022], $E^{(n)}(\mathbf{w}) = \widehat{R}_1(n) - \widehat{R}_3(n)$ is structurally consistent w.h.p. and the result now follows from the previous Lemma 1. □

Remark that to compute the feature $\mathcal{T}_{ij}^{(n)}$ associated with the pair ij and defined in Theorem 1, we only need to process the time series $\{y_i(\ell), y_j(\ell)\}_{\ell=0}^{n}$ associated with this particular pair ij. Namely, note that

$$\mathcal{T}_{ij}^{(n)} := \frac{1}{n} \sum_{\ell=0}^{n-1} \left(y_i(\ell + D) y_j(\ell), \ldots, y_i(\ell + M) y_j(\ell) \right),$$

which only involves the time series of nodes i and j. As such, it is possible to reconstruct the connectivity pattern in a pairwise manner. This *locality* property is not shared with a variety of other estimators. For example, to reconstruct the ij^{th} entry of the Precision matrix or inverse of the empirical covariance matrix $\left(\widehat{R}_0(n) \right)^{-1}$, one needs, in general, to first compute the whole covariance matrix $\widehat{R}_0(n)$. This implies, in particular, that to estimate the ij^{th} entry of the Precision matrix, we need to process the time series of the whole network. If the latter is of a large scale nature, this would be a challenging task on its own.

Definition 4.3 (Identifiability gap for matrix-valued estimators)**.** *Given a matrix-valued estimator $F^{(n)}$, define its* identifiability gap *as*[1]

$$\Gamma\left(F^{(n)}\right) \stackrel{\Delta}{=} \min_{ij\,:\,A_{ij}\neq 0} \mathcal{F}_{ij}^{(n)} - \max_{ij\,:\,A_{ij}=0} \mathcal{F}_{ij}^{(n)}, \tag{4.12}$$

i.e., the gap *between the smallest entry of $\mathcal{F}_{ij}^{(n)}$ across connected pairs and the largest entry of $\mathcal{F}_{ij}^{(n)}$ over disconnected pairs.*

[1]The terminology and definition were borrowed from [Matta et al., 2022] and also used in [Machado et al., 2022].

Observe that a matrix-valued estimator $F^{(n)}$ is structurally consistent w.h.p. if and only if $\Gamma\left(F^{(n)}\right) > 0$ w.h.p., or in other words, if and only if connected pairs are separated from disconnected pairs, in view of the entries of the matrix $\mathcal{F}^{(n)}$, for n large enough, as illustrated in Fig. 4.1. This statistical metric is a relevant parameter regarding the *hardness* of the classification. The larger the identifiability gap, the *easier* the classification via thresholding of the entries of the matrix $\mathcal{F}^{(n)}$ tends to be.

Similarly, we define the identifiability gap $\Gamma\left(T^{(n)}\right)$ associated with a tensor-valued estimator $T^{(n)}$ as the maximum distance among all parallel hyperplanes that consistently separate the features, as in Fig. 4.2, also referred to as *margins*.

Definition 4.4 (Identifiability gap for order-3 tensor-valued estimators)**.** *We define the identifiability gap $\Gamma\left(T^{(n)}\right)$ of a tensor-valued estimator as the distance between the margins as*

$$\Gamma\left(T^{(n)}\right) \triangleq \max_{(\mathbf{w},\tau_1),(\mathbf{w},\tau_2)\in\mathcal{C}} \frac{|\tau_1 - \tau_2|}{||\mathbf{w}||}, \tag{4.13}$$

where $\mathcal{C}$ indexes the set of linear maps that consistently separate the features: $(\mathbf{w},\tau) \in \mathcal{C}$ if and only if the linear map $\mathcal{L}_{\mathbf{w},\tau}(\mathbf{x}) := \mathbf{w}\cdot\mathbf{x} - \tau$ consistently separates the features.

Lemma 2 states that, if one incorporates further structurally consistent matrix-valued estimators in the composition of the feature vector, the associated identifiability gap increases.

Lemma 2. *Let $T^{(n)}$ be a tensor-valued estimator whose underlying features at each pair ij are defined as*

$$\mathcal{T}_{ij}^{(n)} := \left(\left[E_1^{(n)}\right]_{ij}, \left[E_2^{(n)}\right]_{ij}, \ldots, \left[E_M^{(n)}\right]_{ij}\right) \in \mathbb{R}^M, \tag{4.14}$$

with identifiability gap $\Gamma_E^{(n)} \triangleq \Gamma\left(T^{(n)}\right)$. Let $\widehat{A}^{(n)}$ be a matrix-valued estimator with identifiability gap $\Gamma_A^{(n)} \triangleq \Gamma\left(\widehat{A}^{(n)}\right)$. If both $\widehat{A}^{(n)}$ and $T^{(n)}$ are (linearly) structurally consistent w.h.p., then the tensor-valued estimator $\widetilde{T}^{(n)}$ defined as

$$\widetilde{\mathcal{T}}_{ij}^{(n)} := \left(\left[\widehat{A}^{(n)}\right]_{ij}, \left[E_1^{(n)}\right]_{ij}, \ldots, \left[E_M^{(n)}\right]_{ij}\right) \in \mathbb{R}^M, \tag{4.15}$$

exhibits an identifiability gap obeying $\Gamma\left(\widetilde{T}^{(n)}\right) \geq \left|\left|\Gamma^{(n)}\right|\right|_2$ w.h.p., with $\Gamma^{(n)} := \left(\Gamma_A^{(n)}, \Gamma_E^{(n)}\right)$.

Proof. Let $\mathsf{CH}\left(\mathcal{S}\right)$ denote the *convex hull* of a set $\mathcal{S} \subset \mathbb{R}^K$, i.e., the smallest convex set containing $\mathcal{S}$ [Hiriart-Urruty and Lemaréchal, 2001]. Define $\widetilde{\mathcal{C}} \triangleq \left\{\widetilde{\mathcal{T}}_{ij}^{(n)}\right\}_{ij\,:\,A_{ij}=1}$ serving as the set of augmented features associated with connected pairs and $\widetilde{\mathcal{D}} \triangleq \left\{\widetilde{\mathcal{T}}_{ij}^{(n)}\right\}_{ij\,:\,A_{ij}=0}$ associated with disconnected pairs. Similarly, define $\mathcal{C} \triangleq \left\{\mathcal{T}_{ij}^{(n)}\right\}_{ij\,:\,A_{ij}=1}$ and $\mathcal{D} \triangleq \left\{\mathcal{T}_{ij}^{(n)}\right\}_{ij\,:\,A_{ij}=0}$. Let R be the smallest entry of $\widehat{A}^{(n)}$ across

connected pairs and r be the greatest entry of $\widehat{A}^{(n)}$ across disconnected pairs. We have that

$$\begin{aligned}\Gamma\left(\widetilde{T}^{(n)}\right)^2 &= \mathsf{d}\left(\mathsf{CH}\left(\widetilde{\mathcal{C}}\right),\mathsf{CH}\left(\widetilde{\mathcal{D}}\right)\right)^2 \\ &\geq \mathsf{d}\left(\mathsf{CH}\left(\mathcal{C}\times[R,\infty)\right),\mathsf{CH}\left(\mathcal{D}\times(-\infty,r]\right)\right)^2 \\ &\geq \mathsf{d}\left(\mathsf{CH}\left(\mathcal{C}\right),\mathsf{CH}\left(\mathcal{D}\right)\right)^2+(R-r)^2 \\ &= \left(\Gamma_E^{(n)}\right)^2+\left(\Gamma_{\widehat{A}}^{(n)}\right)^2=\left|\left|\Gamma^{(n)}\right|\right|^2\end{aligned},$$

where the first identity conforms to an alternative definition to identifiability gap; and the first inequality holds in view of the inclusions $\mathsf{CH}\left(\widetilde{\mathcal{C}}\right)\subset\mathsf{CH}\left(\mathcal{C}\times[R,\infty)\right)$ and $\mathsf{CH}\left(\widetilde{\mathcal{D}}\right)\subset\mathsf{CH}\left(\mathcal{D}\times(-\infty,r]\right)$. This concludes the proof. □

Lemma 2 asserts that, if further matrix-valued structurally consistent estimators are incorporated into the feature vector, the identifiability gap increases. This result further motivates a paradigm that promotes the pursuit for further structurally consistent matrix-valued estimators: (i) characterize these estimators (a research endeavor on its own); (ii) stack them so to yield an order-3 tensor-valued estimator; (iii) train nonlinear separation schemes (e.g., Convolutional Neural Networks) or linear ones (e.g., SVMs with a linear kernel) to perform estimation. Aiming to boost sample-complexity performance, nonlinear machine learning techniques should be preferred over the linear ones.

4.4 Methodology

In order to stratify the pairs of nodes into connected or disconnected from the observed time series, we address the linear separability property of the covariance-based features $\left\{\mathcal{T}_{ij}^{(n)}\right\}_{ij}$ established in Theorem 1, by studying the performance of trained classifiers, in particular, Support Vector Machines (SVM) with linear kernel and CNNs. CNNs were chosen over other ANNs architectures because (i) they are capable of extracting higher-level (more abstract) features and (ii) reduce the number of parameters by removing less informative features. The training set is given by

$$\mathsf{Tr}^{(n)}\triangleq\left\{\left(\overline{\mathcal{T}}_{ij}^{(n)},\mathbf{1}_{\left\{A_{ij}\neq 0\right\}}\right)\right\}_{i\neq j} \tag{4.16}$$

where we have introduced the normalized feature vectors

$$\overline{\mathcal{T}}_{ij}^{(n)}:=\frac{\mathcal{T}_{ij}^{(n)}}{\max_{i\neq j}\left|\left|\mathcal{T}_{ij}^{(n)}\right|\right|_\infty}, \tag{4.17}$$

with the unnormalized features given by,

$$\mathcal{T}_{ij}^{(n)}\triangleq\left(\left[\widehat{R}_{-100}(n)\right]_{ij},\left[\widehat{R}_{-99}(n)\right]_{ij},\ldots,\left[\widehat{R}_{100}(n)\right]_{ij}\right).$$

In other words, for training, we provide a normalized feature vector $\overline{\mathcal{T}}_{ij}^{(n)}$ associated with the pair ij as input to a classifier and the output should be the ground truth $1_{\{A_{ij}\neq 0\}}$, i.e., whether i links to j or not. We remark that the same data set is used to train the SVM and the CNN classifiers.

The normalization in the training set is motivated by the following observation. When we have infinitely many samples, i.e., $n = \infty$, then

$$\mathcal{T}_{ij}^{\infty} = \sigma^2 \left(\left[\overline{R}_D\right]_{ij}, \left[\overline{R}_{D+1}\right]_{ij}, \ldots, \left[\overline{R}_M\right]_{ij} \right) \tag{4.18}$$

where $\overline{R}_k$ is the k-lag covariance matrix of the normalized process $(\mathbf{y}(n)/\sigma)_{n\in\mathbb{N}}$, i.e., the process whose noise is normalized to unit variance. With the proposed normalization in equation (4.17), the multiplicative factor σ^2 is canceled out, which ideally decreases the role played by the noise-level in the performance of the trained CNNs. Furthermore, this normalization renders the generalization performance of the trained CNNs robust across structurally distinct graphs.

The trained CNN will ideally assume the value 1 for features associated with disconnected pairs and 2 over connected pairs. Therefore, as the estimated values are expected to be close to one of the values of the classes, we simply apply a threshold at 1.5 and the values closer to each class are classified as belonging to that class, as illustrated in Fig. 4.3.

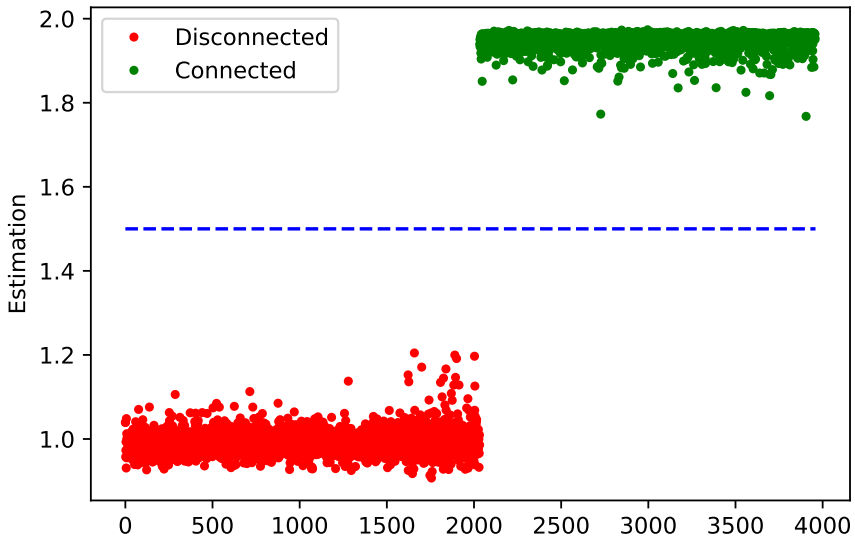


Figure 4.3: CNN thresholding process: entries that are closer to '1' are classified as a disconnected pair whereas those closer to '2' are classified as connected.

To generate the matrix A to obtain the time series data $\{\mathbf{y}(\ell)\}_{\ell=0}^{n}$ following the dynamics (4.1), given a graph G, the following procedure was considered. Let G be a given graph without self-loops, i.e., $G_{ii} = 0$ for all i. Define the interaction

matrix A as

$$\begin{cases} A_{ij} &= \alpha_1 \frac{G_{ij}}{d_{\max}(G)}, & \text{for } i \neq j \\ A_{ii} &= \alpha - \sum_{k \neq i} A_{ik}, & \text{for all } i \end{cases}, \tag{4.19}$$

where $d_{\max}(G)$ is the maximum *in-flow* degree of the underlying graph G and $0 < \alpha_1 \leq \alpha < 1$ are some constants. In other words, the rows of A sum up to $\alpha < 1$ and its support graph is given by G. This is often cast as the *Laplacian rule* [Sayed, 2014]. The interaction matrix thus yields a stable networked dynamical system (4.1) and with a support graph of interactions given by G. To generate G, we considered the realization of random graph models as Erdős–Rényi random graph models for undirected graphs, binomial random graph models for directed graphs, and also real-world networks.

Regarding the CNN training process, the data set is partitioned into three data sets: 70% of the samples are reserved for training, 15% for validation and the remaining 15% for testing. The CNN architecture, illustrated in Fig. 4.4, contains three series of 1D convolution layers that promote the extraction of higher-level features with ReLU activation function interleaved by max pooling layers which down sample the parameters. After Max pooling is performed, features are fed into two fully connected layers which approximate the data to one of the classes. During training, the weights of the network are updated according to Adam algorithm, an alternative to the classical stochastic gradient descent which computes individual adaptive learning rates for different parameters from estimates of first and second moments of the gradients. To evaluate the difference between the current estimation value and the ground truth, we adopted the mean squared error loss function. To achieve the best performing model, the loss function is monitored at each epoch of training and, when the loss function reaches a minimum (after a number of epochs with no improvement), the training stops. A functionally designated as early stopping. This method allows us to specify a large number of training epochs and stop training once the model performance stops improving on a hold out validation dataset. That happens because we use many epochs, which may lead to overfitting, and few epochs result in underfitting of the model.

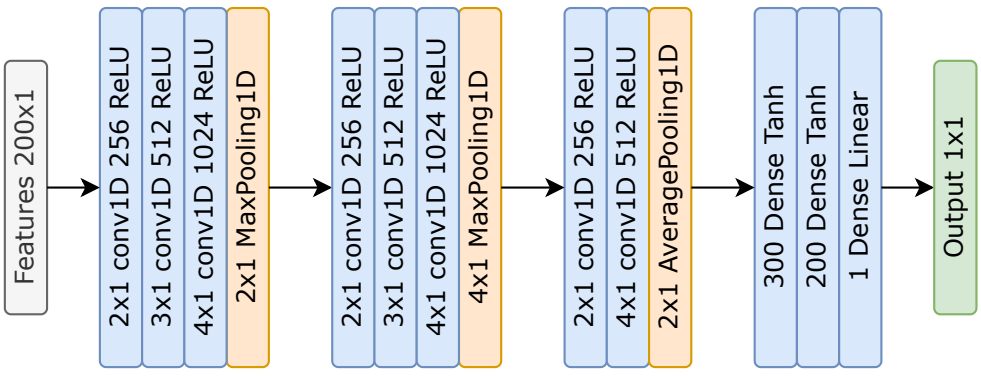


Figure 4.4: CNN architecture.

In order to achieve the best model, different CNNs are trained over the same data set (with different runs) and the one with the highest performance on the

testing set is chosen. The metric adopted is the identifiability gap between the estimations (or predictions) of disconnected and connected pairs.

Chapter 5

Simulation Results and Validation

In this section, we present the numerical results validating the proposed CNN-based methodology and we compare our algorithm with other state-of-the-art graph learning methods. To evaluate the performance of the proposed approach, we focus on three main metrics: (i) *accuracy* – represents the likelihood of correct graph recovery; (ii) *identifiability gap* – represents the separability between the two clusters (associated with connected and disconnected pairs); and (iii) *cluster variance* – attempts to model the *tightness* of the clusters. These metrics will be described in detail in the respective sections (some have already been introduced in the previous chapter). These are relevant metrics for classification tasks. Remark that the main goal of a graph learning algorithm is to recover the underlying graph with the least possible amount of time series samples, i.e., to have the best possible recovery *accuracy* with the minimal amount of samples. This goes by saying that the algorithms should minimize *sample-complexity*. This is often the case when the underlying algorithm yields a *large* identifiability gap and a *small* cluster variance. Indeed, the large separability between clusters and the tightness of the clusters render the task of automatic separation of the underlying features (and thus, consistent classification) *easier* as illustrated in Fig. 5.1. For these reasons, we look closely at the dependence of these metrics with respect to the number of time series samples for the distinct algorithms.

Each metric is plotted against the number of time series samples n. Since we are recovering the connectivity graph under partial observability, we will assume a limited number of observable nodes (we set this parameter to $|S| = 20$ observable nodes), regardless of the size of the actual graph. We consider 1000 Monte Carlo runs for each experiment and plot the average of those runs.

The comparison was carried out among the following methods

- The trained CNN over our covariance-based features;
- The trained SVM with a linear kernel over our covariance-based features;
- Granger under partial observability $\left[\widehat{R}_1(n)\right]_{\mathcal{S}} \left(\left[\widehat{R}_0(n)\right]_{\mathcal{S}}\right)^{-1}$ which is structurally consistent [Matta et al., 2020, 2022; Santos et al., 2020b] for distinct regimes of graph connectivity (dense or sparse);

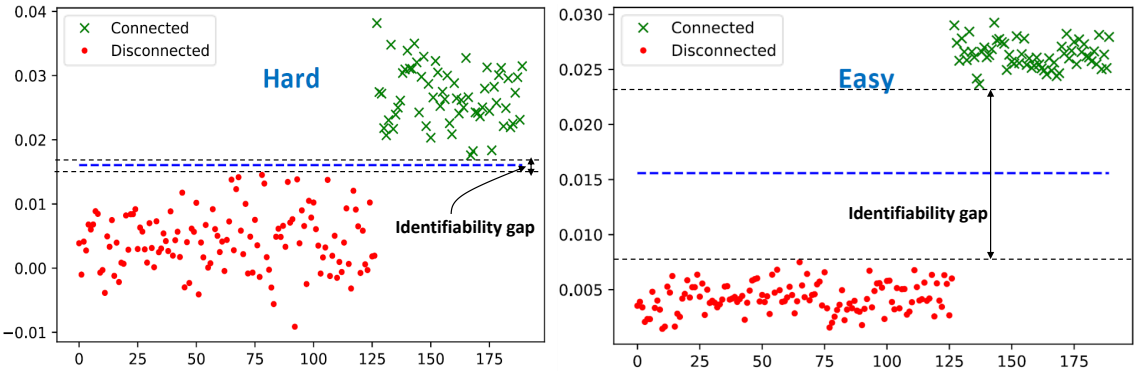


Figure 5.1: Example illustrating the *hardness* of the classification problem. The plots depict the output of two matrix-valued estimators: each point in the abscissa represents a pair of nodes in the graph and in the ordinate are the values assigned by the corresponding estimator to each of these pairs. The pairs are sorted in the abscissa so that disconnected pairs lie on the left (of 125) and connected pairs lie on the right (for visualization purposes). The red dots represent the values assigned to disconnected pairs and the green crosses are for connected pairs. It is harder to automatically set the right threshold to consistently classify the pairs on the left plot as the identifiability gap is smaller and the clusters are wider. That is, the estimator on the right will tend to produce better results.

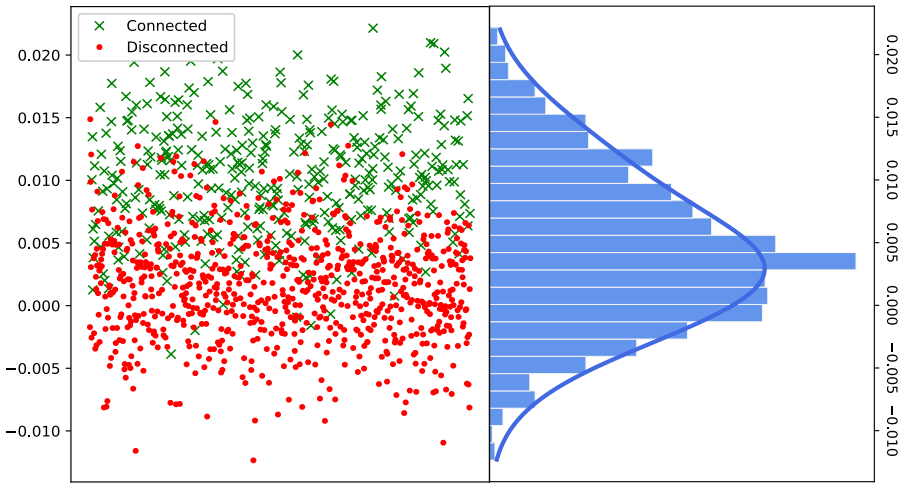
- The one-lag estimator $\widehat{R}_1(n)$, which is also consistent for several graph connectivity regimes [Matta et al., 2022];
- The $\widehat{R}_1(n) - \widehat{R}_3(n)$ that is structurally consistent [Chen et al., 2022] regardless of the connectivity pattern.

The choice for the comparison with these particular algorithms is because these methods are known to be state-of-the-art algorithms over a great range of connectivity (dense or sparse networks) and they are probably consistent under partial observability (our focus) for linear stochastic networked dynamical systems [Chen et al., 2022; Matta et al., 2022].

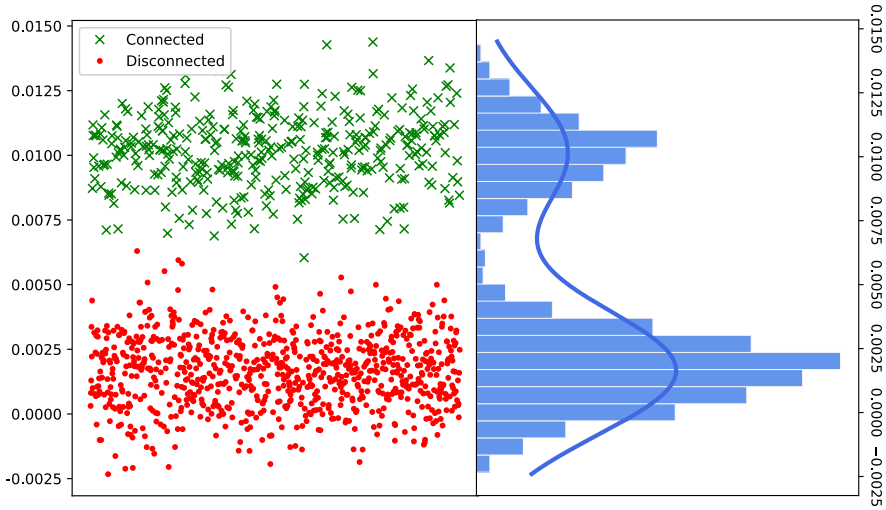
Remark that while the SVM provides an automatic classification of the features (as associated to either connected or disconnected pairs), and the trained CNN exhibit output values close to either '1' or '2' (i.e., it provides approximately an automatic classification for the features) where '1' represents the label of disconnected pairs and '2' represents the label for connected pairs, the latter three (matrix-valued) estimators require an extra postprocessing of its matrix entries to be clustered into two groups (a group associated with connected pairs and a group associated with disconnected pairs) as illustrated in Fig. 5.2. For these matrix-valued estimators, we apply Gaussian Mixture Model (GMM) over the sorted entries of the matrix-valued estimators in order to consistently stratify the pairs into connected or disconnected. This was proposed in [Chen et al., 2022].

The choice of this method is further motivated by the following properties: (i) no need to specify initial parameters beyond the number of clusters (which are two in our framework); (ii) stable under clusters of quite distinct sizes (which in our framework will depend on the graph connectivity); and (iii) capable of fitting

ellipsoid shapes, formed by the matrix-estimator entries when there is a positive identifiability gap as presented in Fig. 5.2. There are various methods that fulfill the first criteria, some of the most used being the hierarchical clustering (both agglomerative and divisive algorithms), K-means or GMM. However, it is known that hierarchical clustering performs badly when dealing with clusters of different sizes and K-means is not engineered to fit ellipsoid shaped data. For these reasons and motivated by its successful use in [Chen et al., 2022], we have chosen GMM to postprocess the matrix-valued estimators. The GMM algorithm is implemented by the expectation-maximization algorithm, i.e., it performs maximum likelihood estimation in the presence of latent variables, which is ideal considering the distribution of the entries when the time-series is large enough, as displayed in Fig. 5.2(b).



(a) $N = 200$, $p = 0.3$, $n = 5e4$



(b) $N = 200$, $p = 0.3$, $n = 5e5$

Figure 5.2: Scatter plots and histograms of the entries of the Granger Estimator for an undirected realization of the Erdős–Rényi random model with $N = 200$ nodes and probability of edge drawing $p = 0.3$. The number of samples is given by n.

Random graph models. To generate the underlying graphs of the networked dynamical systems in order to obtain the time series samples, we resort to the realization of two well-known random graph models. In particular, an edge at a pair of nodes ij is placed with probability p. In other words, $\mathbb{P}\left(G_{ij}=1\right)=p$, for each pair ij, where $G_{ij}\in\{0,1\}$ is the ij^{th} entry of the adjacency matrix G representing the graph. Further, the entries of G are independent Bernoulli random variables. To generate undirected graphs, we simply generate the upper triangular part of the matrix G following the random realization just described and force symmetry, i.e., $G_{ij}=G_{ji}$ (this is known as an Erdős–Rényi model). For directed networks, we perform the experiment over all entries of G (independently) without enforcing symmetry (this is known as Binomial random graph).

Generative model and training. The time series data is generated following the discrete-time linear dynamics in equation (4.1) over distinct realizations of Erdős–Rényi (for undirected graphs) or Binomial (for directed graphs) random graph models and real-world networks. For the experiments over synthetic interaction graphs, after building the graph as the realization of a random graph model, we build the interaction matrix A on top of it (i.e., the support of the interaction matrix A gives the underlying graph) as discussed around equation (4.19) in Chapter 3. This is a standard policy to assign weights to a graph that is often referred to as *Laplacian rule* [Sayed, 2014]. In what follows, the parameters N and p stand for the number of nodes and probability of edge/arrow drawing in the random graph model. It should be mentioned that the CNN and SVM are trained over a single realization of an Erdős–Rényi random graph (for undirected networks) with $p=0.5$ and $N=100$. Nevertheless, and as we will demonstrate, the CNNs generalize well for different graphs (of distinct sizes and connectivity patterns), either synthetically generated or from real-world scenarios. The choice of the number of nodes N to train the CNN was arbitrary, but the choice $p=0.5$ for the probability of arrow drawing is simply due to its symmetry: it yields a graph with approximately the same number of connected and disconnected pairs, and thus, it does not favor dense or sparse networks in the training.

More on training and generalization. Instead of training the CNNs over a single realization of an Erdős–Rényi random graph model with $N=100$ and $p=0.5$, we could have chosen to train them over several networked dynamical systems generated with distinct random graph models, i.e., generated with distinct N and p. However, our main goal was to demonstrate the generalization property of the CNNs: training them over a single synthetic network (generated with $N=100$ and $p=0.5$) yields remarkable performance across a great range of connectivity patterns and networks of distinct sizes. This generalization property is particularly important as, in general, we might not know the underlying nature of the target networked system where we are attempting to perform causal inference. Additionally, to demonstrate that the CNN-based approach is capable of generalizing for different levels of noise variance σ^2 (i.e., the variance of each entry of the vector $\mathbf{x}(n)$ in the dynamics (4.1)), the value used to generate the training data is $\sigma=0.1$ while for testing is usually $\sigma=0.5$. We will also show in the next section, that the CNNs generalize well across a whole range of noise variance, while trained with $\sigma=0.1$ (refer to Fig. 5.3).

Outline. In Section 5.1, we demonstrate the robustness of the CNN-based model against changes in excitation noise variance in the linear networked dynamical system. Sections 5.2, 5.3 and 5.4 convey the results comparing the estimators over distinct realizations of Erdős–Rényi and Binomial graphs regarding the *accuracy*, *identifiability gap* and *cluster variance*, respectively. Section 5.5 shows the results for two real-world graphs. Section 5.6 presents results suggesting that the inclusion of structurally consistent matrix-valued estimators in the feature vectors tends to further enhance the performance of the CNN-based approach, i.e., improve accuracy with less time series samples. This is particularly motivated by Lemma 2 in Chapter 3. Finally, Section 5.5 observes that higher order lag-moment matrices may convey important structural information in a sense to be made precise. Somehow, this goes against the common wisdom, where it is generally believed that lower-lag covariance matrices are the critical building blocks for graph learning.

We remark that the results presented in Figures 5.6(c), 5.8(a), 5.9(a) and 5.22(b) were already reported in [Machado et al., 2022].

5.1 Robustness against Noise Variance

We start by looking at the stability of the proposed covariance-based feature vectors approach against a great range of excitation noise variance in the networked dynamical system. This property is in part due to the normalization of the covariance-based feature vectors, which also scales the entries of the feature vectors to values between 0 and 1 independently of the network size and connectivity as discussed around equation (4.18) in Chapter 3. Experiments were performed by measuring the accuracy of the trained CNNs and linear SVMs under the same conditions in terms of the underlying interaction graph but applying them against different noise variances in the networked dynamical system.

The results displayed in Fig. 5.3 shows that the plots associated with the CNN and SVM performance are not sensitive to the noise level even though they were trained with $\sigma = 0.1$. Additionally, CNN exhibits better accuracy. This demonstrates that our proposed method based on the training over covariance-based features is robust to different levels of noise.

5.2 Accuracy

Let $\widehat{G}(n)$ be an estimator for the underlying graph computed from n observed time series samples, i.e., if $\widehat{G}_{ij}(n) = 1$, then, the estimator asserts that there is an arrow or edge from node i to node j; otherwise, if $\widehat{G}_{ij}(n) = 0$, then it asserts that there is no arrow from i to j. Let G be the ground-truth graph, i.e., $G_{ij} = 1$ implies that there is actually an edge from i to j and if $G_{ij} = 0$, then there is no direct edge. If $\widehat{G}_{ij}(n) = G_{ij}$, then the estimator $\widehat{G}$ predicted consistently the link from node i to node j from the n observed time series data.

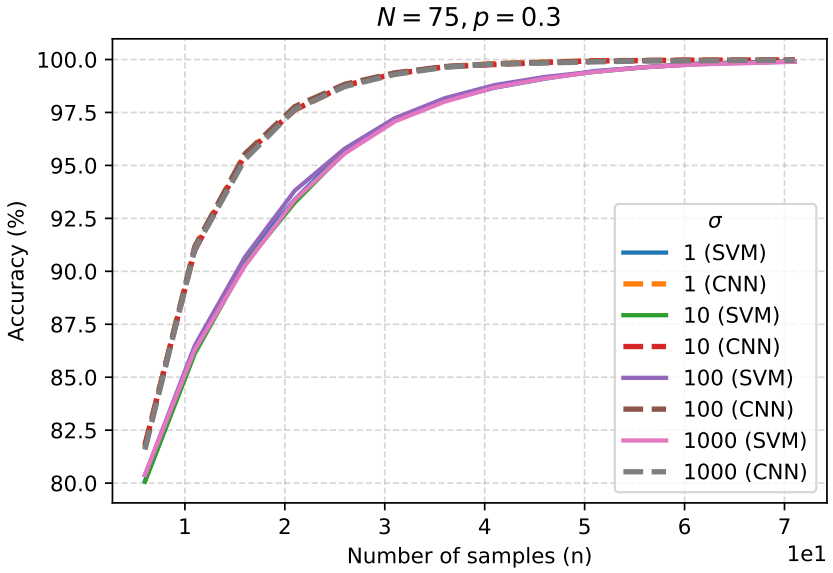


Figure 5.3: The plots illustrate the stability of the CNN and linear SVM classifiers trained over the covariance-based features against distinct noise level regimes. While the CNN and SVM are trained with a time series data under $\sigma = 0.1$, the plots show that they generalize well over other variance regimes. The dashed plots depict the accuracy as a function of the number of time series samples n for the trained CNNs, while continuous plots depict the corresponding performance for the trained SVMs.

We define *accuracy* of an estimator $\widehat{G}$ as the percentage of edges or links correctly classified[1] by this estimator, i.e.,

$$Accuracy \triangleq \frac{\sum_{i=1}^{|S|} \sum_{j \neq i} \mathbf{1}_{\{\widehat{G}_{ij}(n) = G_{ij}\}}}{|S|\,(|S|-1)} \times 100\%, \tag{5.1}$$

where $|S|$ is the number of observed nodes. Remark that when the underlying graph is undirected, then we can simply refer to connected or disconnected pairs and the accuracy reduces to

$$Accuracy \triangleq \frac{C_{\text{pairs}}}{T_{\text{pairs}}} \times 100\%, \tag{5.2}$$

where C_{pairs} is the number of correctly classified pairs (as connected or not) and $T_{\text{pairs}} = |S|\,(|S|-1)\,/2$ is the total number of pairs.

Next, we compare the sample-complexity performance among the considered estimators, i.e., the accuracy as a function of the number of time series samples. The experiments were performed over distinct graphs, with distinct sizes and connectivity regimes, for undirected and directed graphs. Figures 5.4, 5.6 and 5.8 refer to undirected Erdős–Rényi models whereas 5.5, 5.7 and 5.9 refer to directed Erdős–Rényi graphs (also known as Binomial random graphs). The parametric

[1] We exclude self-loops, i.e., links that depart from a node i to itself.

details of the generative random graph model, namely given by N and p, are displayed at the top of each figure.

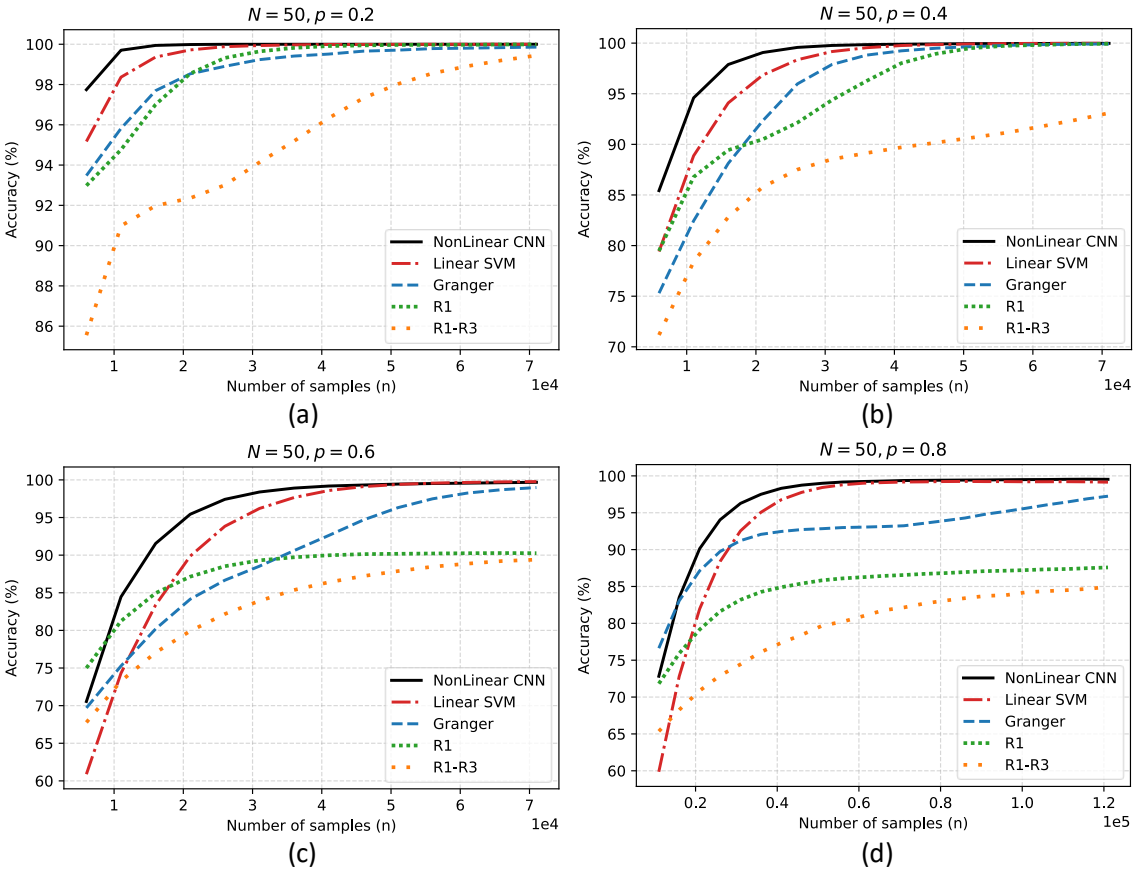


Figure 5.4: Structure estimation performance of the considered algorithms when the underlying support graph of interactions is undirected and drawn from an Erdős–Rényi random graph model with $N = 50$ for distinct regimes of connectivity captured by the probability of edge drawing p.

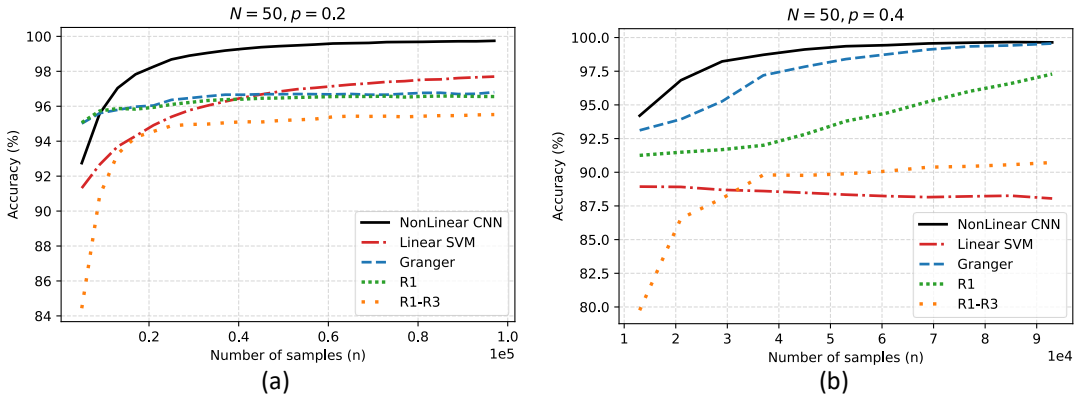


Figure 5.5: Structure estimation performance of the considered algorithms when the underlying support graph of interactions is directed and drawn from a directed Erdős–Rényi random graph model (a.k.a. Binomial random graph model) with $N = 50$ for distinct regimes of connectivity captured by the probability of edge drawing p.

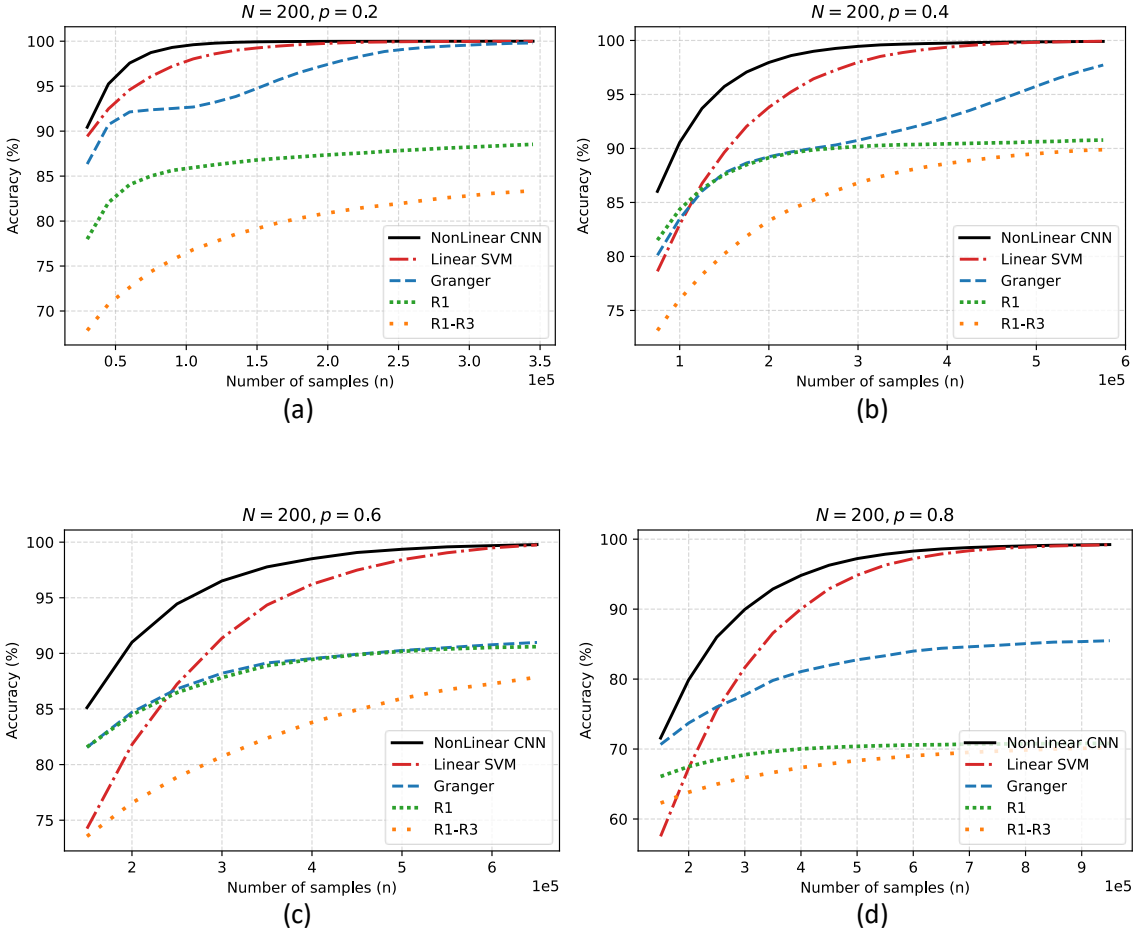


Figure 5.6: Structure estimation performance of the considered algorithms when the underlying support graph of interactions is undirected and drawn from an Erdős–Rényi random graph model with $N = 200$ for distinct regimes of connectivity captured by the probability of edge drawing p.

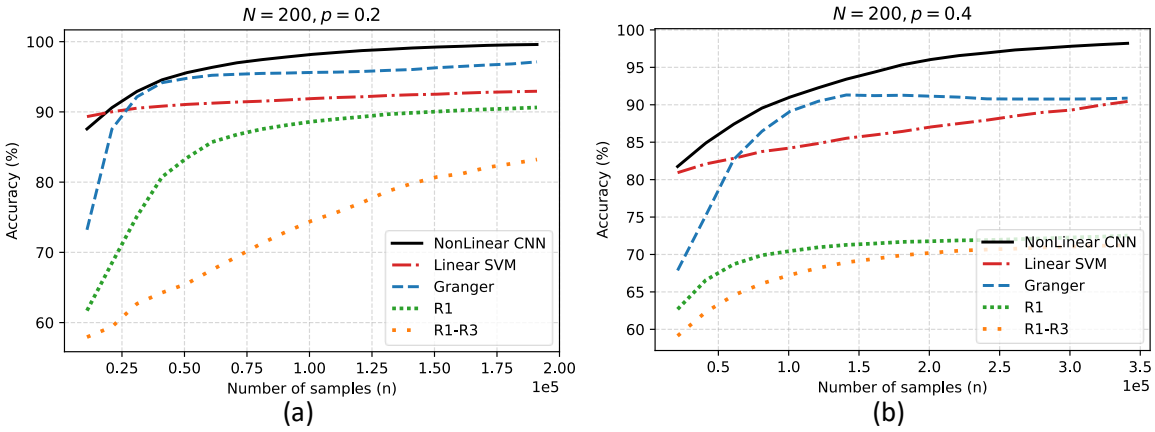


Figure 5.7: Structure estimation performance of the considered algorithms when the underlying support graph of interactions is directed and drawn from a directed Erdős–Rényi random graph model (a.k.a. Binomial random graph model) with $N = 200$ for distinct regimes of connectivity captured by the probability of edge drawing p.

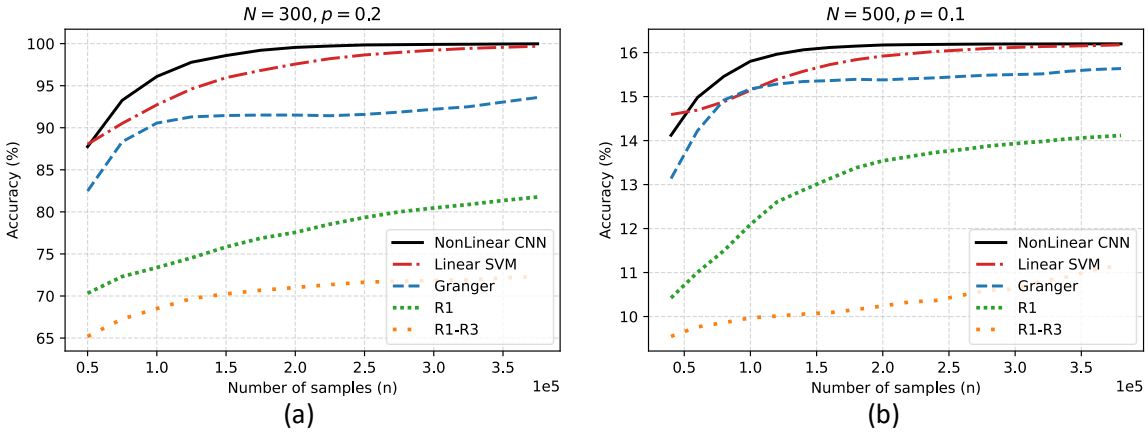


Figure 5.8: Structure estimation performance of the considered algorithms when the underlying support graph of interactions is undirected and drawn from an Erdős–Rényi random graph model with $N = 300$ and $N = 500$ for distinct regimes of connectivity captured by the probability of edge drawing p.

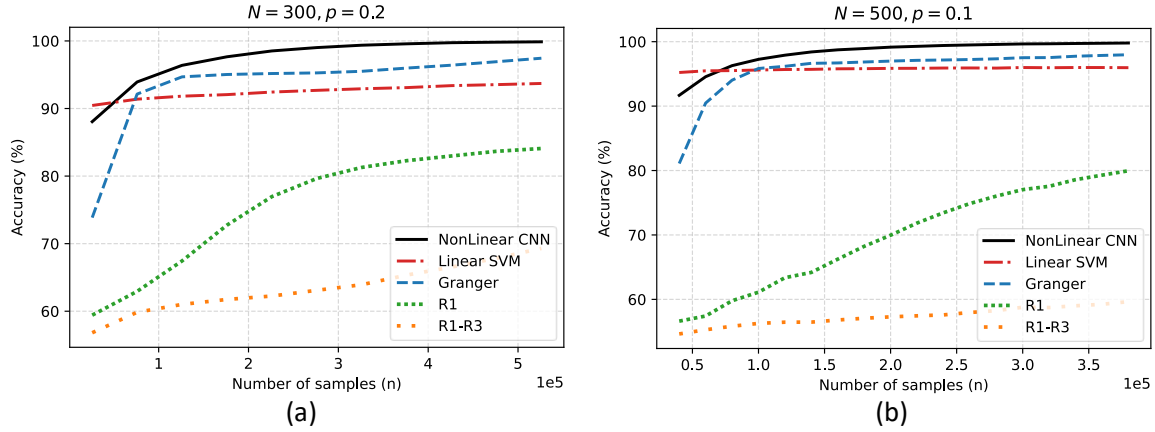


Figure 5.9: Structure estimation performance of the considered algorithms when the underlying support graph of interactions is directed and drawn from a directed Erdős–Rényi random graph model (a.k.a. Binomial random graph model) with $N = 300$ and $N = 500$ for distinct regimes of connectivity captured by the probability of edge drawing p.

The results show the overall superiority in sample-complexity performance for the CNN-based classifier. In every test scenario, the CNN-based approach was able to consistently recover the connectivity of the observable sub-graph with a lower number of samples (smaller time-series), when compared with the matrix-valued estimators and the linear SVM.

5.3 Identifiability Gap

The identifiability gap, introduced in equation (4.12), measures the difference between the minimum value entry across connected pairs and the maximum value

entry over connected pairs. As previously mentioned, this statistical metric is an essential indicator that may provide a preview for the estimator's performance. We performed several experiments to depict the dependence of the identifiability gap with the number of samples n for each of the estimators. We remark that these experiments are made under the same environment conditions as the accuracy experiments, in terms of the Erdős–Rényi parameterization. Note that the linear SVM automatically classifies the features as belonging to a certain class (disconnected or connected), therefore it is not included in the following charts. Also, estimators entries are normalized between '0' and '1' so the gap between clusters is measured on the same scale.

Figures 5.10, 5.12 and 5.14 refer to experiments on undirected graphs whereas 5.11, 5.13 and 5.15 refer to experiments on directed graphs. The results obtained on the dependence of the identifiability gap with the number of samples n are consistent with the behavior observed in the accuracy charts, namely, estimators with higher identifiability gap tend to exhibit better accuracy performance. This confirms the direct relationship between accuracy performance and the size of the gap between clusters yielded by the estimators. The CNN-based approach which presented the best accuracy performance (as presented in the previous subsection) is also the estimator capable of attaining the biggest gap between clusters with the least amount of time series samples. There is also an important difference in comparison with the matrix-valued estimators, which exhibit smaller identifiability gap values. The accuracy of the $\widehat{R}_1(n) - \widehat{R}_3(n)$ and $\widehat{R}_1(n)$ matrix-valued estimators is consistent with their smaller identifiability gaps exhibited.

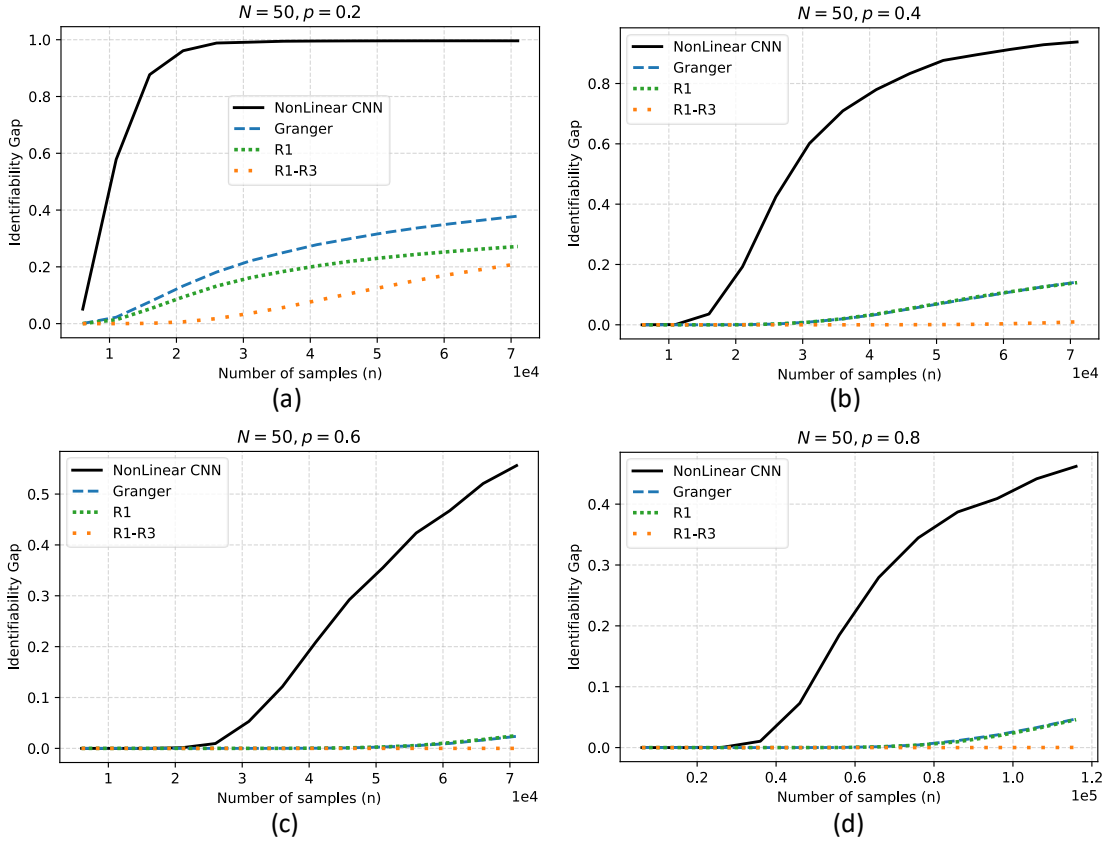


Figure 5.10: Identifiability gap as a function of the number of time series samples for the distinct estimators over undirected graphs with $N = 50$ and for distinct regimes of connectivity given by p.

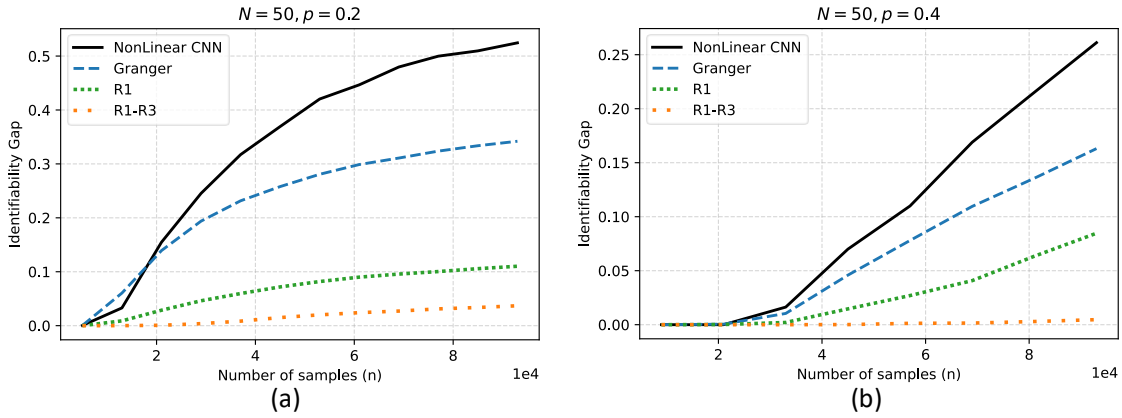


Figure 5.11: Identifiability gap as a function of the number of time series samples for the distinct estimators over directed graphs with $N = 50$ and for distinct regimes of connectivity given by p.

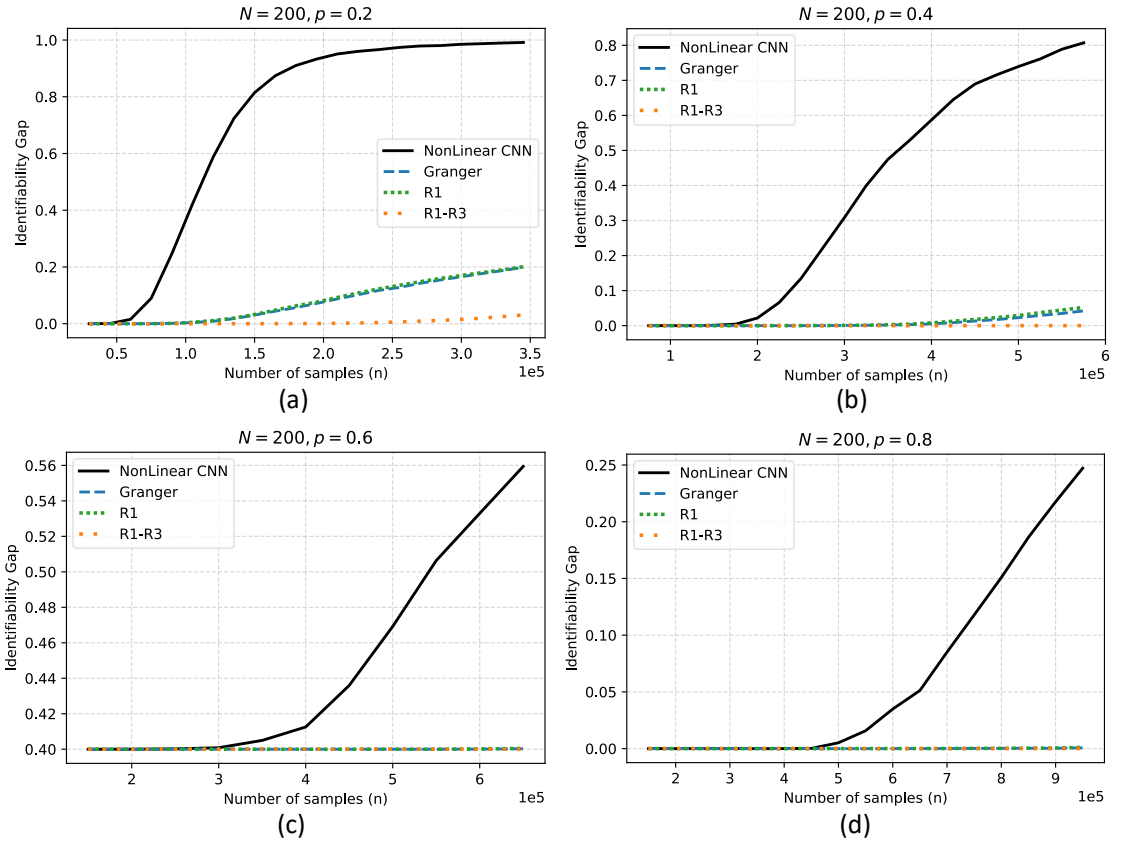


Figure 5.12: Identifiability gap as a function of the number of time series samples for the distinct estimators over undirected graphs with $N = 200$ and for distinct regimes of connectivity given by p.

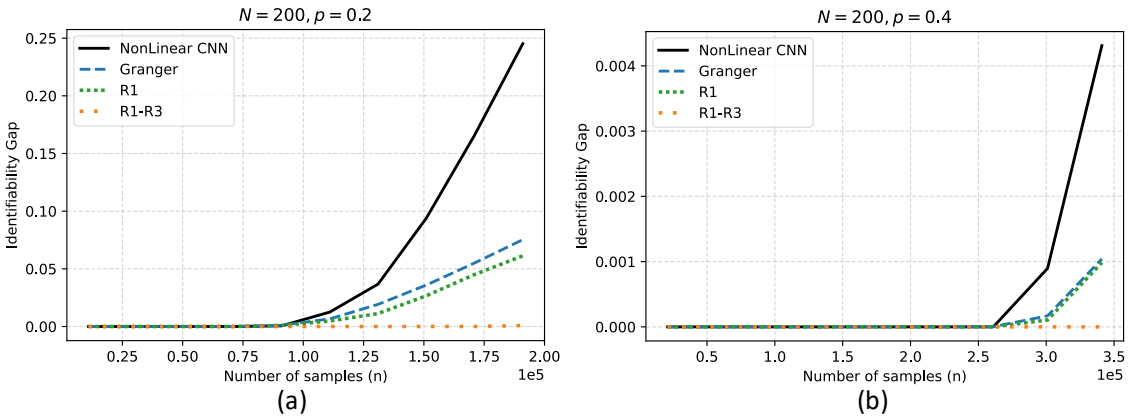


Figure 5.13: Identifiability gap as a function of the number of time series samples for the distinct estimators over directed graphs with $N = 200$ and for distinct regimes of connectivity given by p.

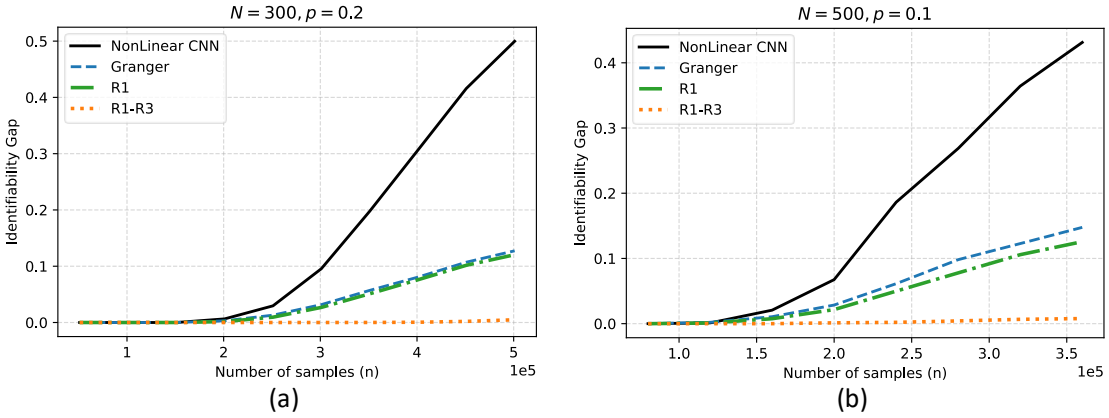


Figure 5.14: Identifiability gap as a function of the number of time series samples for the distinct estimators over undirected graphs with $N = 200$ and $N = 500$.

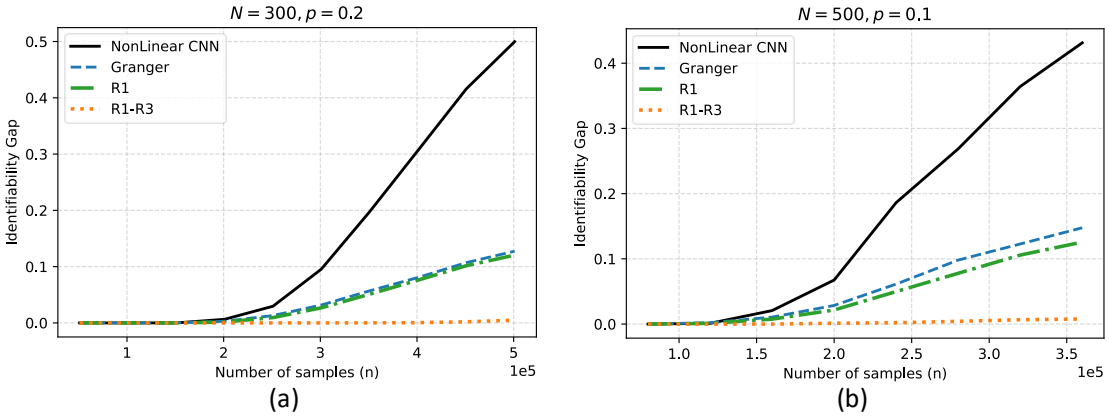


Figure 5.15: Identifiability gap as a function of the number of time series samples for the distinct estimators over directed graphs with $N = 200$ and $N = 500$.

5.4 Clusters Variance

The *clusters variance* is defined as the average between the empirical variance over the entries of each cluster (the one for connected pairs of nodes and the one for disconnected pairs). First, let $\widehat{A}$ be a matrix-valued estimator, and $\widehat{A}_{ij}(n)$ be the value assigned to the link from node i to node j by this estimator computed from n time series samples. We formally define

$$\begin{aligned} \mu_c &= \frac{\sum_{i\neq j\,:\,i\to j}\widehat{A}_{ij}(n)}{|C|} \\ C_{\mathsf{var}} &= \frac{\sum_{i\neq j\,:\,i\to j}(\widehat{A}_{ij}(n)-\mu_c)^2}{|C|} \end{aligned} \tag{5.3}$$

where $|C|$ represents the number of links or arrows in the graph, μ_c is the mean of the cluster of connected pairs, C_{var} is the variance of the cluster of connected pairs.

Similarly, we define the cluster of disconnected pairs.

$$\begin{aligned} \mu_d &= \frac{\sum_{i\neq j\,:\,i\not\to j}\widehat{A}_{ij}(n)}{|D|} \\ D_{\mathsf{var}} &= \frac{\sum_{i\neq j\,:\,i\not\to j}(\widehat{A}_{ij}(n)-\mu_d)^2}{|D|} \end{aligned} \tag{5.4}$$

where $|D|$ is the number of disconnected pairs.

Finally, we define the cluster's variance as the average $Cl\_var \triangleq (C_{var}+D_{var})/2$.

The cluster's variance conforms to a measure of the *tightness* of the clusters. Similarly to the experiments on the identifiability gap, the SVM will not be included in the following charts, as it automatically classifies the features as belonging to a certain class (disconnected or connected). The estimator's entries are also normalized. Further, the experiments were performed under the same conditions in terms of the underlying random graph.

Figs. 5.16, 5.18 and 5.20 refer to experiments on undirected graphs whereas Figures 5.17, 5.19 and 5.21 refer to directed graphs. Our results show that matrix-valued estimators have a slow decrease of the cluster variance and slow convergence with the increase of the number of samples. Still, the $\widehat{R}_1(n)-\widehat{R}_3(n)$ estimator has slightly more scattered clusters. Regarding the CNN-based approach results, overall, it exhibits clusters with greater variance over a reduced number of samples, but quickly converges to a small or even zero cluster variance. We remark, that the CNN starts with a higher variance because it is trained to fit (or rather approximate) one of two classes, therefore with insufficient time-series samples wrong estimations become more likely which increases this measure, whereas matrix-valued estimators have the entries overall dispersed as previously illustrated in Fig. 5.2.

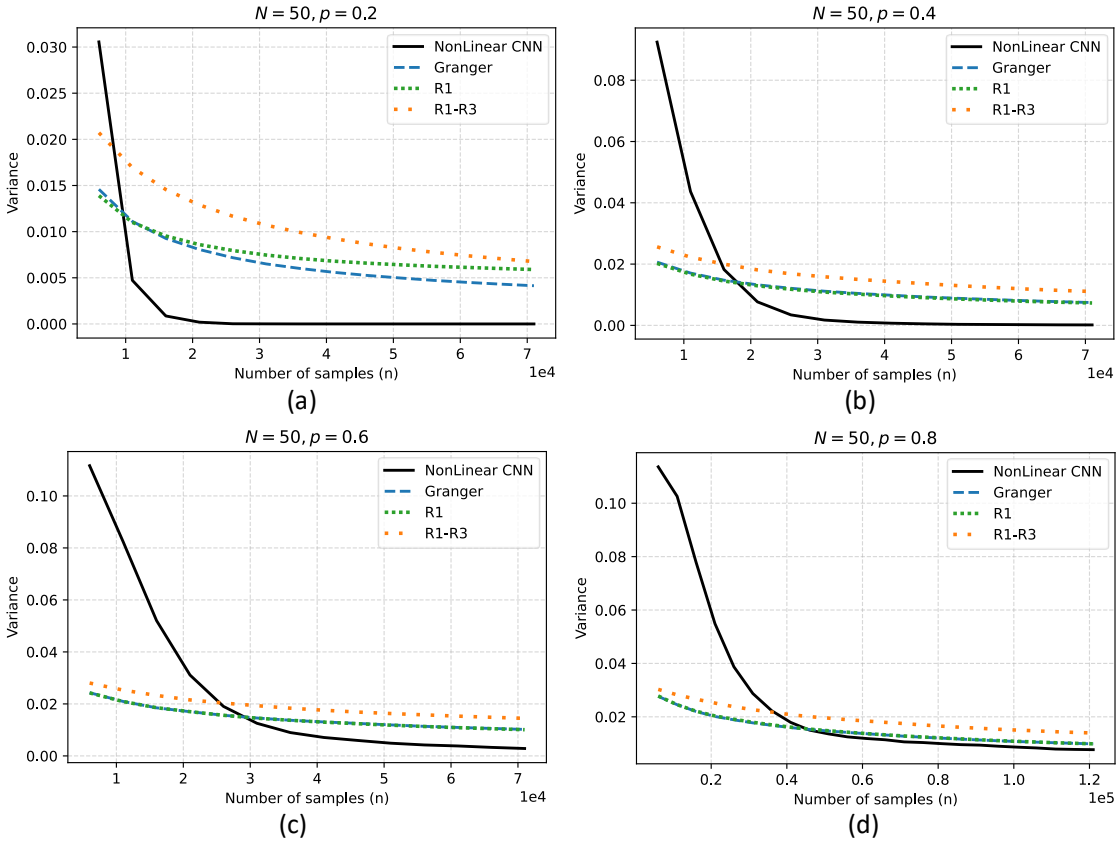


Figure 5.16: Clusters variance of undirected graphs generated via the Erdős–Rényi model with $N = 50$ and distinct regimes of connectivity given by p.

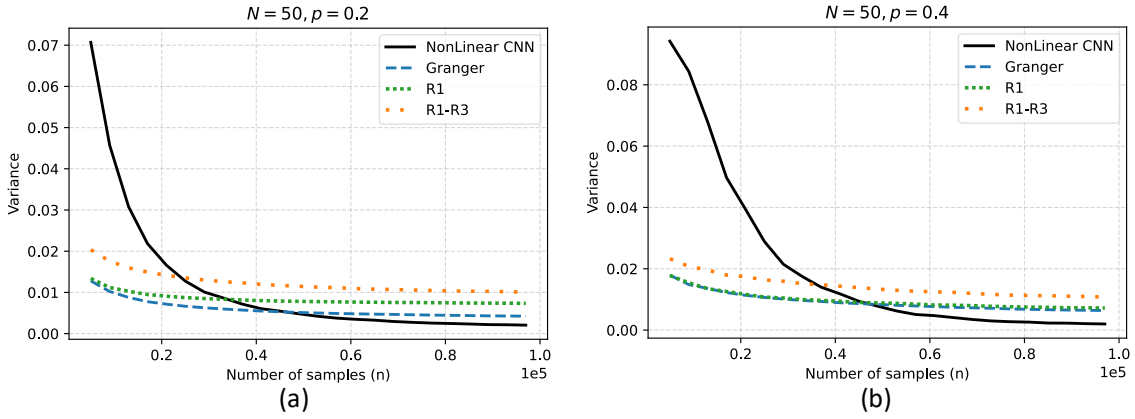


Figure 5.17: Clusters variance of directed graphs generated via the Binomial random model with $N = 50$ and distinct regimes of connectivity given by p.

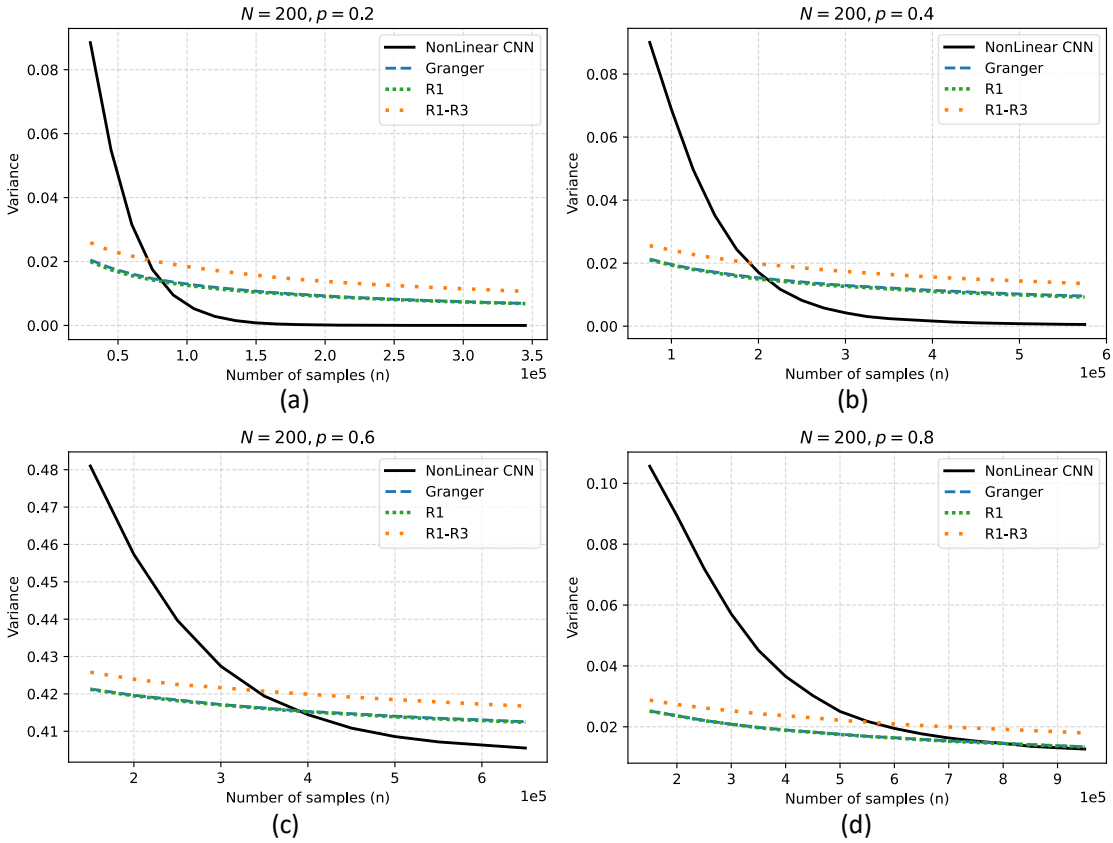


Figure 5.18: Clusters variance of undirected graphs generated via the Erdős–Rényi model with $N = 200$ and distinct regimes of connectivity given by p.

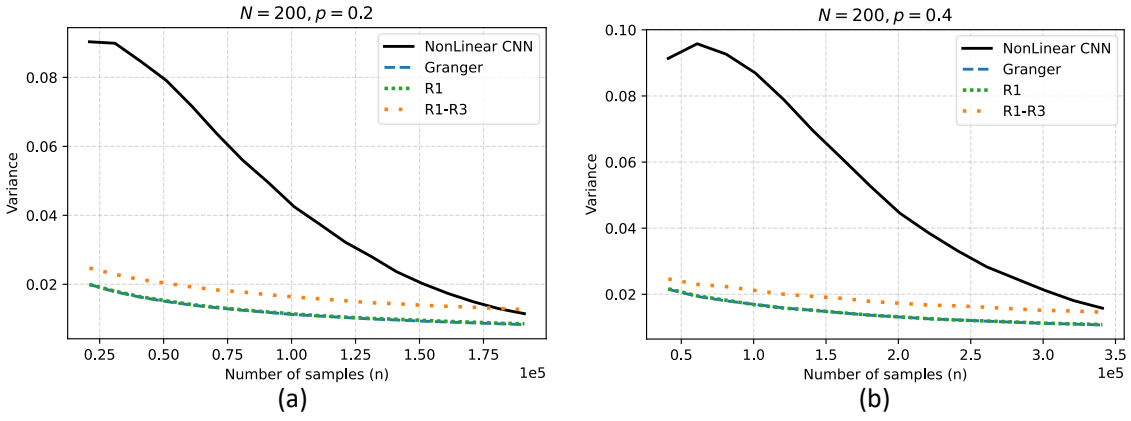


Figure 5.19: Clusters variance of directed graphs generated via the Binomial random model with $N = 200$ and distinct regimes of connectivity given by p.

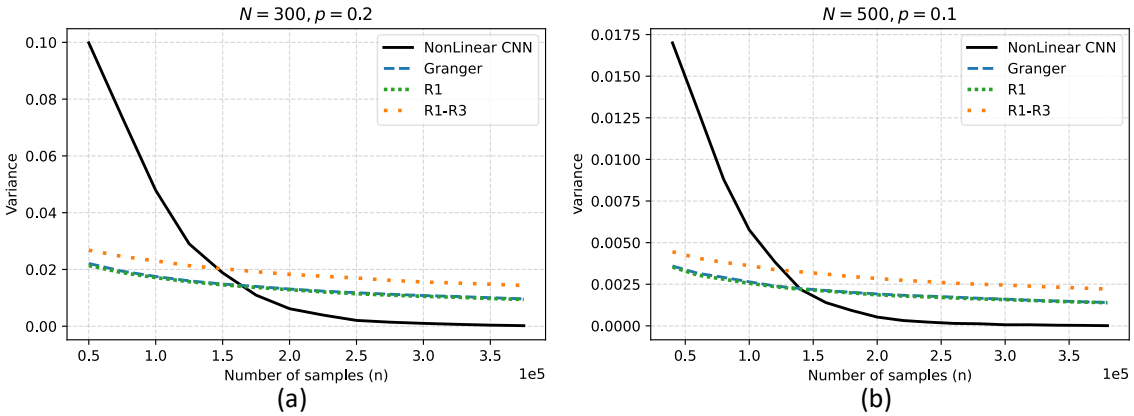


Figure 5.20: Clusters variance of undirected graphs generated via the Erdős–Rényi model with $N = 200$ and $N = 500$.

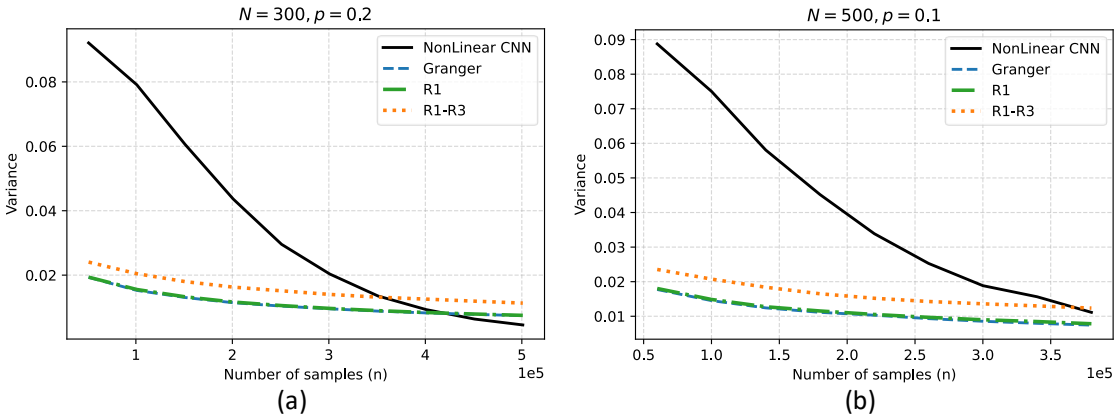


Figure 5.21: Clusters variance of directed graphs generated via the Binomial random model with $N = 200$ and $N = 500$.

5.5 Real-world Networks

In this section, we present results on real-world networks. The main goal is to demonstrate the generalization property of the method: while the CNNs are trained over features computed from the time series data of a linear networked dynamical system with a connectivity pattern given by a single realization of an Erdős–Rényi with $N = 100$ and $p = 0.5$, the method performs well not only over other realizations of Erdős–Rényi random graphs (with distinct connectivities p and sizes N), but also over real-world networks. We tested the estimator's performance on recovering the graph connectivity over real-world networks under partial observability with the same linear dynamics, obtained from the public repository [Rossi and Ahmed, 2015].

We tested two real-world graphs of distinct areas. First (a) represents the brain structural connectivity matrix of a monkey and (b) represents an enzyme biochemical network. Here, we measured sample-complexity performance, i.e., the accuracy as a function of the number of time series samples n similarly to what

was done in Section 5.2. The recovery of the graph connectivity is extremely important, specially for brain networks, as it would describe interactions between distinct parts of the brain. Note that both the linear SVM and the CNN-based approaches were trained over a single realization of an Erdős–Rényi random graph model and applied to these real-world networks. A way of improving these results would be to specialize the training of the CNNs to examples provided by each framework (e.g., Brain networks, enzyme networks, etc.), but the main focus on this thesis was to demonstrate how the method generalizes.

Results show that the CNN-based approach is capable of generalizing well and outperforms the matrix-based estimators for these graphs, particularly for the monkey brain network as illustrated in Fig. 5.22. For the enzyme biochemical network, as illustrated in 5.23, remarkably, the SVM exhibits an overall superior performance over the CNN.

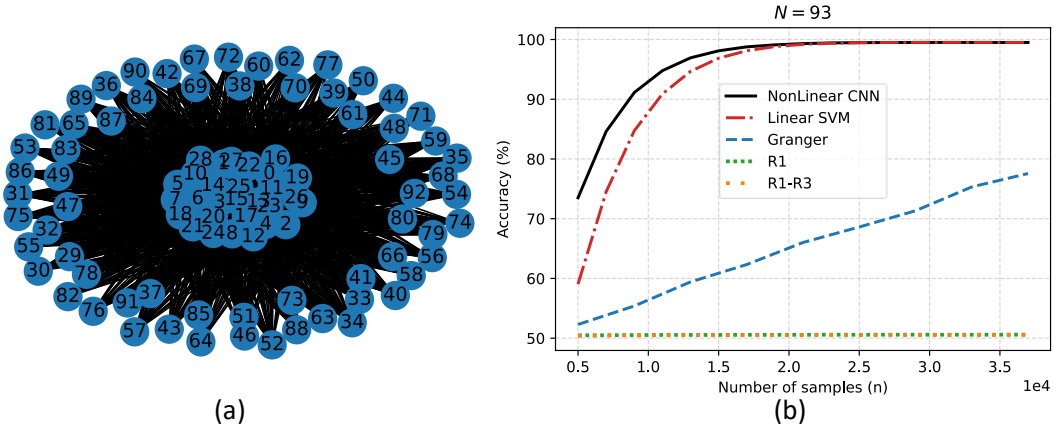


Figure 5.22: Structure estimation performance for the brain structural connectivity matrix of a monkey.

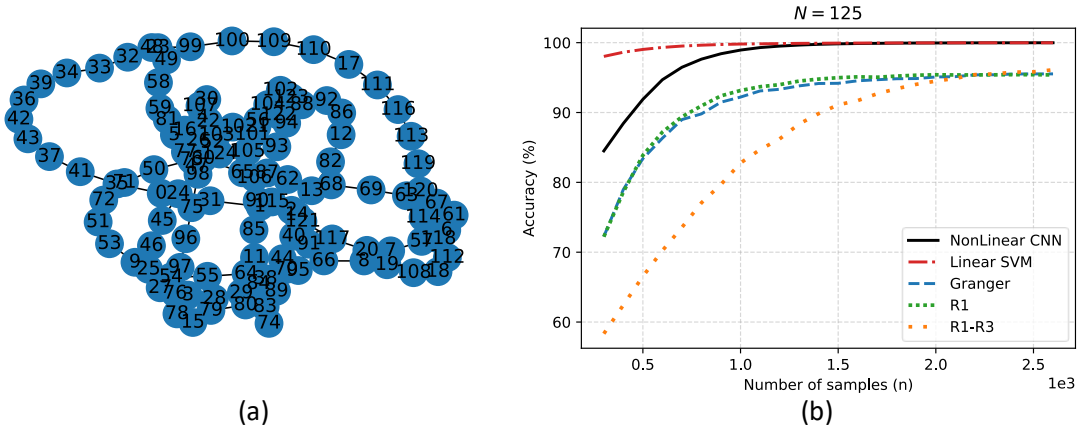


Figure 5.23: Structure estimation performance for an enzyme biochemical network.

5.6 Incorporation of New Features

Lemma 2, in Chapter 3, asserts that the inclusion of further structurally-consistent matrix-valued estimators in the feature-vector increases the identifiability gap (in feature space), i.e., the *separability* between the clusters of features associated with connected and disconnected pairs. Given that the Granger estimator is the one with best performance overall, we test its inclusion in the covariance-based features, namely, we consider

$$\mathcal{T}_{ij}^{(n)} \triangleq \left(\left[\widehat{A}_{\mathcal{S}} \right]_{ij}, \left[\widehat{R}_{-100}(n) \right]_{ij}, \ldots, \left[\widehat{R}_{100}(n) \right]_{ij} \right)$$

where the additional component $\widehat{A}_{\mathcal{S}} \triangleq \left[\widehat{R}_1(n) \right]_{\mathcal{S}} \left(\left[\widehat{R}_0(n) \right]_{\mathcal{S}} \right)^{-1}$ is the Granger under partial observability, with only $|S| = 20$ nodes observed.

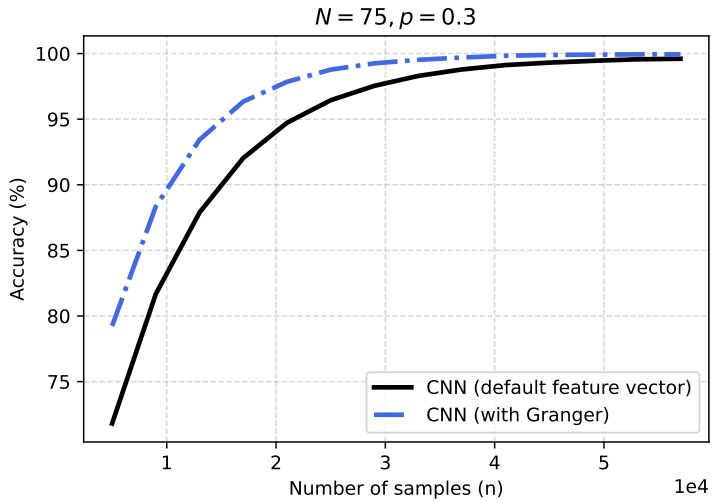


Figure 5.24: Inclusion of the Granger estimator in the feature vector.

Figure 5.24 shows that the inclusion of the Granger estimator in the feature-vector significantly improves the sample-complexity performance, i.e., it is able to completely recover the underlying graph connectivity with a lower number of samples. This motivates the further study of structurally consistent matrix-valued estimators to build new sets of features with greater separability properties and thus, yielding better performance.

5.7 High Order Lag-moments also Convey Relevant Structural Information

While the matrix-valued estimators referred so far (and comprising the bulk of the literature) are built primarily on *low lag-moments*, i.e., on covariance matrices $\widehat{R}_k(n)$ where k is small enough, we illustrate in this section that, remark-

ably, higher-order lag-moments may convey important information for consistent graph learning from the observed time series samples. To be more concrete about the relevance or not of lag-moments, we mean the following: Let $\mathcal{L}_{\mathbf{w},\tau}(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} - \tau$ be a linear map that consistently separates the features $\{\mathcal{T}_{ij}\}_{ij}$, in that $\mathcal{L}_{\mathbf{w},\tau}(\mathcal{T}_{ij}) > 0$ if the node i links to j and $\mathcal{L}_{\mathbf{w},\tau}(\mathcal{T}_{ij}) \leq 0$, otherwise. Note that w_ℓ is the weight assigned to the ℓ^{th} lag-moment $\widehat{R}_\ell(n)$. Thus, if the weight vector $\mathbf{w}$ underlying the separating linear map $\mathcal{L}_{\mathbf{w},\tau}$ assumes high values at the higher-moment entries of the feature vector, then the higher moments must be conveying most of the structural information.

To study this question, we look at the linear map obtained by an SVM with linear Kernel. Fig. 5.25 depicts the weights of the vector $\mathbf{w}$ across the lags. This experiment actually confirms that low lag moments like $\widehat{R}_0, \widehat{R}_1, \widehat{R}_2$ and $\widehat{R}_3$, conveys most of the structural information as the higher weights were concentrated about these lower lag covariance matrices.

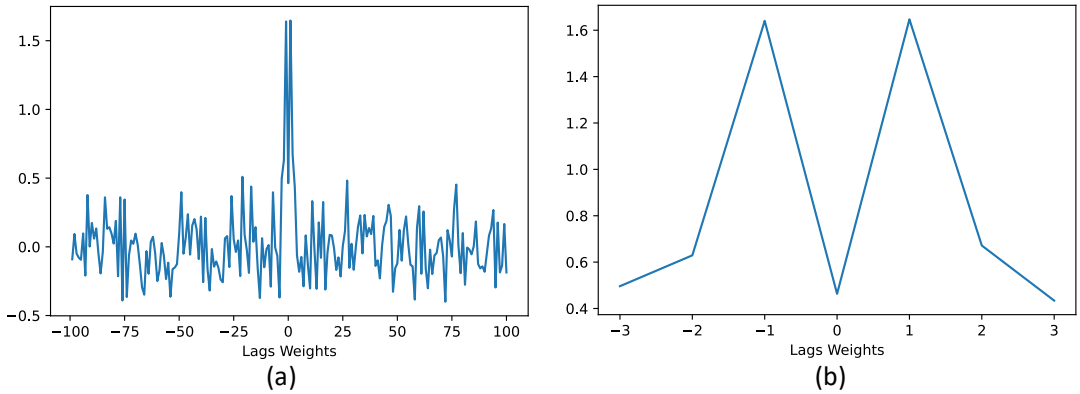


Figure 5.25: (a) represents the weights attained by the SVM; (b) zooms on the first three lags.

However, we can inquire whether there are other separating linear maps $\mathcal{L}_{\mathbf{w},\tau}$ which assign larger weights to higher-order lag-moments. For this, we resort to another linear classifier. We choose to train a Feedforward Neural Network (FFNN) with linear activation functions. The vector of weights $\mathbf{w}$ of the resulting separating linear map is depicted in 5.26.

Surprisingly, it bestows low weights upon low lag-moments and higher ones over higher moments. This also illustrates that distinct linear classifiers may display distinct behavior while exhibiting good performance. For the performance of the linear SVM and FFNN underlying these experiments, please refer to Fig. 5.27.

In our particular experiment, we recall that the SVM and the FFNN resort to distinct optimization strategies, the SVM finds the separating maximum-margin hyperplanes that consistently separate the clusters. On the other hand, the FFNN finds the linear approximation underlying the *true* classification function with the optimization over its weights being governed by the stochastic gradient descent algorithm.

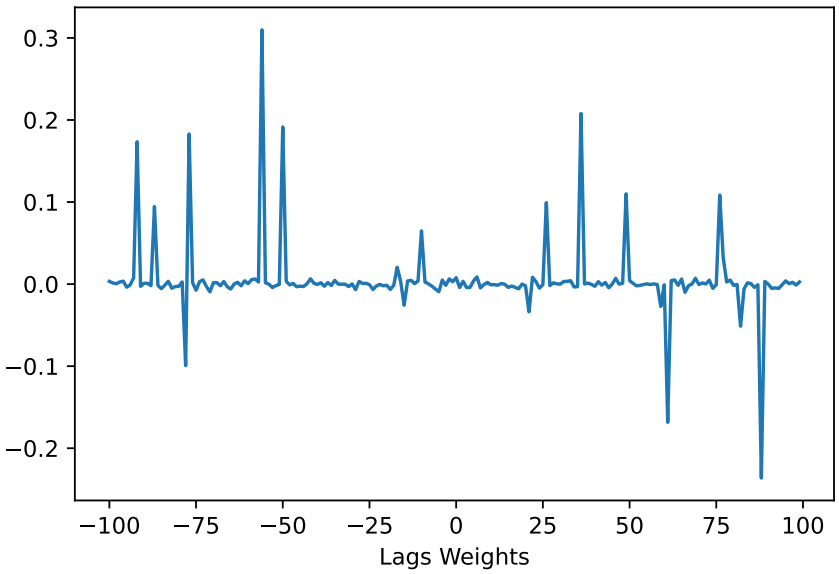


Figure 5.26: Weights achieved by the FFNN after training.

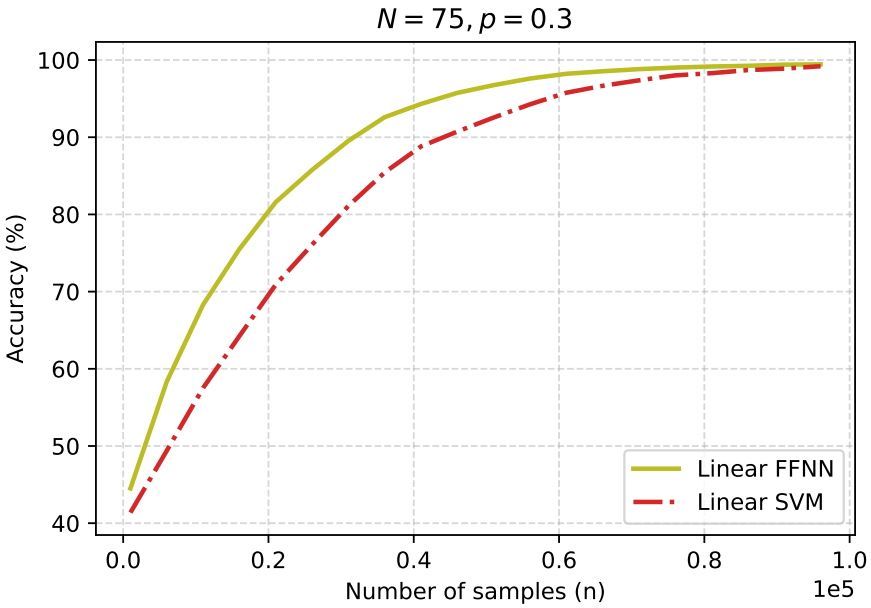


Figure 5.27: Accuracy vs number of time series samples associated with the linear SVM and the FFNN with linear activation functions trained over the covariance-based features.

Chapter 6

Concluding Remarks

In this section, we summarize the main contributions of this thesis: we revisit the chart of proposed goals (presented at the beginning of the semester) to state clearly what has been accomplished, and we point to future directions of research in the area of graph learning of networked dynamical systems.

6.1 Contributions

This thesis addressed the problem of learning the graph of interactions from the observed time series data stemming from linear stochastic networked dynamical systems under partial observability, i.e., the time series data of only a subset of nodes are available. In particular, we proposed a novel set of feature-vectors computed from the available time series as statistical descriptors for the connectivity of each pair of nodes. We proved that the set of features is linearly separable (with high probability), that is, there exists a hyperplane (in feature space) that separates the features associated with connected pairs of nodes from the features associated with disconnected pairs of nodes. Therefore, if we have the right separating hyperplane, we can consistently classify the pairs as connected or not. In particular, distinct machine learning methods can be trained over these features. We have chosen to train CNNs over this set of features to obtain a state-of-the-art causal inference method. The trained CNNs exhibited remarkable generalization: while trained over a particular synthetic network (obtained via the realization of an Erdős–Rényi random graph model with $N = 100$ and $p = 0.5$) it performs well over a whole range of connectivity regimes including real-world networks. We have also observed that, contrary to common wisdom, higher-order lag-moments can convey important structural information. Some results have been submitted for publication and other results are in preparation.

6.2 Proposed vs accomplished goals

In January 2022, we presented the proposal summarized in the Gantt chart, Fig. 6.1, for the goals to be pursued in the second semester. Next, we state what has been accomplished, point-by-point.

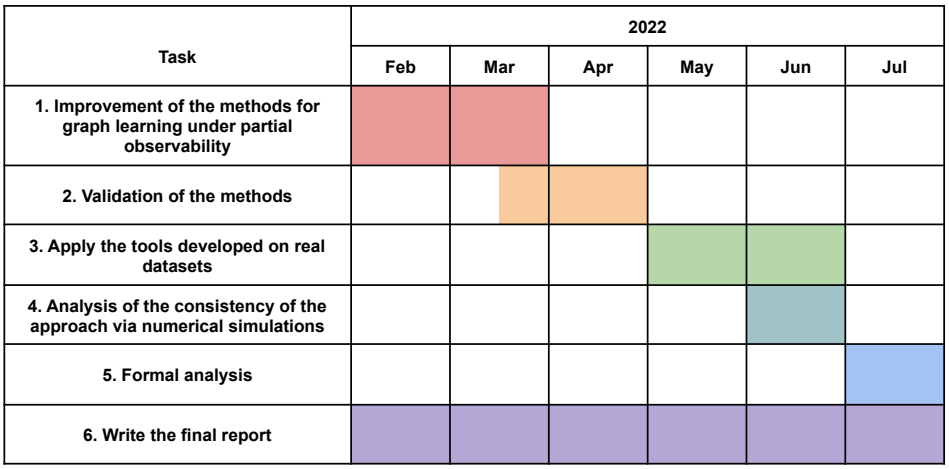


Figure 6.1: Distribution of the tasks for the second semester of 2022.

Point 1**: Improvement of the methods for graph learning under partial observability.** Indeed, a state of the art graph learning method has been proposed. The algorithm is tailored to the partial observability setting in that the network can be reconstructed in a pairwise manner, i.e., the algorithm consistently decides whether there is an arrow or edge at each pair of nodes from observation of only the time series data of the pair.

Point 2**: Validation of the methods.** In Chapter 4, we presented an extensive amount of simulations demonstrating that not only the underlying interaction graph can be reconstructed from the available time series data, but it exhibits competitive sample-complexity performance.

Point 3**: Apply the results developed on real data sets.** In Chapter 4, we presented results over real-world networks.

Point 4**: Analysis of the consistency of the approach via numerical simulations.** As referred in Point 1, extensive simulations demonstrated the structural consistency of the approach, namely, that the graph can be faithfully inferred if sufficient time series data is provided.

Point 5**: Formal analysis.** In Chapter 3, we presented results that formally proved the linear separability of the proposed features. Remark that establishing some form of separability (even if nonlinear) is a necessary condition to grant structural consistency of the method. Moreover, we proved that the incorporation of further structurally consistent matrix-valued estimators into the feature vector tends to enhance the separability properties of the features and thus, improve the performance of the method. The formal results were submitted for publication [Machado et al., 2022].

6.3 Future directions

Several open questions remain. While our causal inference method applies successfully to directed graphs, the formal results were established (in Chapter 3) for undirected graphs and assuming that the covariance matrix of the excitation noise is diagonal, i.e., a multiple of the identity matrix. Ongoing work is being pursued towards addressing the directed networks case as well as the colored noise framework, where the underlying covariance matrix of the noise term $\mathbf{x}(n)$ assumes a more general form.

Further, the feature-vector approach, as a statistical descriptor for the connectivity of each pair, detoured from the overall graph learning approach in the literature. The majority of works consider assigning real-value estimates for the coupling strength between nodes. This said, the work developed suggests that new feature-vectors with perhaps better separability and tightness properties could be designed. For instance, Lemma 2 suggests that the pursuit for novel matrix-valued estimators could be useful to build new feature-vectors.

Lastly, the approach proposed applies indirectly to a broad class of nonlinear systems. In particular, those nonlinear dynamical systems exhibiting the equilibrium as a global attractor. This is because, at the equilibrium (and with a small enough noise level) the system behaves approximately as a linear dynamical system. However, methods that leverage directly on the nonlinear form of the vector-field describing nonlinear networked dynamical systems could exhibit better performance. In particular, feature-vectors designed specifically for these systems could boost performance over them.

Proof. Since $E^{(n)}(\mathbf{w}) = \sum_{\ell=1}^{M} w_\ell E_\ell^{(n)}$ is structurally consistent w.h.p. for some $\mathbf{w} \in \mathbb{R}^M$, then there exists a threshold $\tau_{\mathbf{w}}$ so that $\left[E^{(n)}(\mathbf{w})\right]_{ij} > \tau_{\mathbf{w}}$ across connected pairs ij and $\left[E^{(n)}(\mathbf{w})\right]_{ij} < \tau_{\mathbf{w}}$, otherwise. Therefore, the affine map $\mathcal{L}_{\mathbf{w}}(\mathbf{x}) = \mathbf{x} \cdot \mathbf{w} - \tau_{\mathbf{w}}$ consistently separates the set of features $\left\{\mathcal{T}_{ij}^{(n)}\right\}_{ij}$ w.h.p. Indeed,

$$\mathcal{L}_{\mathbf{w}}(\mathcal{T}_{ij}^{(n)}) = \mathcal{T}_{ij}^{(n)} \cdot \mathbf{w} - \tau_{\mathbf{w}} = [E(\mathbf{w})]_{ij} - \tau_{\mathbf{w}} > 0 \tag{6.1}$$

for a connected pair ij or

$$\mathcal{L}_{\mathbf{w}}(\mathcal{T}_{ij}^{(n)}) = \left[E^{(n)}(\mathbf{w})\right]_{ij} - \tau_{\mathbf{w}} < 0, \tag{6.2}$$

otherwise. In other words, the hyperplane characterized by the linear map $\mathcal{L}_{\mathbf{w}} : \mathbb{R}^M \longrightarrow \mathbb{R}$ separates consistently the pairs ij for all $i \neq j$. □

References

Jeffrey Adams, Niels Richard Hansen, and Kun Zhang. Identification of partially observed causal models: Graphical conditions for the linear non-gaussian and heterogeneous cases. In *Advances in Neural Information Processing Systems 34 pre-proceedings (NeurIPS 2021)*, NeurIPS '21, 2021.

Charu C. Aggarwal. An introduction to neural networks. In *Neural Networks and Deep Learning*, pages 1–52. Springer International Publishing, 2018.

Hamed Habibi Aghdam. *Guide to convolutional neural networks*. Springer International Publishing, Cham, Switzerland, 1 edition, May 2017.

Timo Ahonen, Abdenour Hadid, and Matti Pietikäinen. Face description with local binary patterns: application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):2037–2041, dec 2006. doi: 10.1109/tpami.2006.244.

Kathleen T. Alligood, Tim D. Sauer, and James A. Yorke. *Chaos*. Textbooks in Mathematical Sciences. Springer, New York, NY, September 2000.

Animashree Anandkumar and Ragupathyraj Valluvan. Learning loopy graphical models with latent variables: Efficient methods and guarantees. *Ann. Statist.*, 41(2):401–435, 04 2013. doi: 10.1214/12-AOS1070.

Animashree Anandkumar, Vincent Y. F. Tan, Furong Huang, and Alan S. Willsky. High-dimensional gaussian graphical model selection: Walk summability and local separation criterion. *J. Mach. Learn. Res.*, 13(1):2293–2337, August 2012.

Animashree Anandkumar, Daniel Hsu, Adel Javanmard, and Sham Kakade. Learning linear bayesian networks with latent variables. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 249–257, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

Roy M. Anderson and Robert M. May. Directly transmitted infections diseases: Control by vaccination. *Science*, 215(4536):1053–1060, 1982. doi: 10.1126/science.7063839.

Nicolas Bacaër. McKendrick and kermack on epidemic modelling (1926–1927). In *A Short History of Mathematical Population Dynamics*, pages 89–96. Springer London, 2011.

Albert-Laszlo Barabasi and Reka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, October 1999.

Laurent Baratchart, Monique Chyba, and Jean-Baptiste Pomet. A grobman–hartman theorem for control systems. *Journal of Dynamics and Differential Equations*, 19(1):75–107, jul 2006. doi: 10.1007/s10884-006-9014-5.

Marya Bazzi, Mason A. Porter, Stacy Williams, Mark McDonald, Daniel J. Fenn, and Sam D. Howison. Community detection in temporal multilayer networks, with an application to correlation networks. 2015. doi: 10.48550/ARXIV.1501.00040.

José Bento, Morteza Ibrahimi, and Andrea Montanari. Learning networks of stochastic differential equations. November 2010.

Andrej Bogdanov, Elchanan Mossel, and Salil Vadhan. The complexity of distinguishing markov random fields. In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, pages 331–342, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

Fred Brauer. Mathematical epidemiology: Past, present, and future. *Infect. Dis. Model.*, 2(2):113–127, May 2017.

Fred Brauer and Carlos Castillo-Chavez. *Mathematical Models in Population Biology and Epidemiology*. Springer New York, 2012. doi: 10.1007/978-1-4614-1686-9.

Alfredo Braunstein, Luca Dall'Asta, Guilhem Semerjian, and Lenka Zdeborová. Network dismantling. *Proceedings of the National Academy of Sciences*, 113(44):12368–12373, 2016. doi: 10.1073/pnas.1605083113.

Guy Bresler, David Gamarnik, and Devavrat Shah. Hardness of parameter estimation in graphical models. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'14, pages 1062–1070, Cambridge, MA, USA, 2014. MIT Press.

Tom Britton. Basic stochastic transmission models and their inference, 2018.

Tom Britton. Epidemic models on social networks—with inference. *Statistica Neerlandica*, 74(3):222–241, February 2020.

Tom Britton, Mathias Lindholm, and Tatyana Turova. A dynamic network in a dynamic population asymptotic properties. *Journal of Applied Probability*, 48(4):1163–1178, 2011.

Elizabeth Bruch and Jon Atwell. Agent-based models in empirical social research. *Sociol. Methods Res.*, 44(2):186–221, May 2015a.

Elizabeth Bruch and Jon Atwell. Agent-based models in empirical social research. *Sociol. Methods Res.*, 44(2):186–221, May 2015b.

José M. Carcione, Juan E. Santos, Claudio Bagaini, and Jing Ba. A simulation of a COVID-19 epidemic based on a deterministic SEIR model. *Frontiers in Public Health*, 8, May 2020.

Venkat Chandrasekaran, Pablo A. Parrilo, and Alan S. Willsky. Latent variable graphical model selection via convex optimization. *Ann. Statist.*, 40(4):1935–1967, 08 2012. doi: 10.1214/11-AOS949.

Yupeng Chen, Zhiguo Wang, and Xiaojing Shen. An unbiased symmetric matrix estimator for topology inference under partial observability. *IEEE Signal Processing Letters*, 29(02):1257–1261, 2022.

David Maxwell Chickering. Optimal structure identification with greedy search. *J. Mach. Learn. Res.*, 3(null):507–554, mar 2003. doi: 10.1162/153244303321897717.

Emily S. C. Ching and H. C. Tam. Reconstructing links in directed networks from noisy dynamics. *Phys. Rev. E*, 95:010301, Jan 2017.

Diego Colombo, Marloes H. Maathuis, Markus Kalisch, and Thomas S. Richardson. Learning high-dimensional directed acyclic graphs with latent and selection variables. *The Annals of Statistics*, 40(1):294–321, 2012.

Ivan Nunes da Silva, Danilo Hernane Spatti, Rogerio Andrade Flauzino, Luisa Helena Bartocci Liboni, and Silas Franco dos Reis Alves. *Artificial Neural Networks*. Springer International Publishing, 2017. doi: 10.1007/978-3-319-43162-8.

Junaid Farooq and Mohammad Abid Bazaz. A novel adaptive deep learning model of covid-19 with focus on mortality reduction strategies. *Chaos, Solitons & Fractals*, 138:110148, September 2020.

Philipp Geiger, Kun Zhang, Bernhard Schoelkopf, Mingming Gong, and Dominik Janzing. Causal inference by identification of vector autoregressive processes with hidden components. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, pages 1917–1925. PMLR, 07–09 Jul 2015.

Pavel Golik, Patrick Doetsch, and Hermann Ney. Cross-entropy vs. squared error training: a theoretical and experimental comparison. In *Interspeech 2013*. ISCA, August 2013.

C. W. J. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37(3):424–438, 1969.

Johannes Günther, Patrick M. Pilarski, Gerhard Helfrich, Hao Shen, and Klaus Diepold. First steps towards an intelligent laser welding architecture using deep neural networks and reinforcement learning. *Procedia Technology*, 15:474–483, 2014. doi: https://doi.org/10.1016/j.protcy.2014.09.007. 2nd International Conference on System-Integrated Intelligence: Challenges for Product and Production Engineering.

Shaobo He, Yuexi Peng, and Kehui Sun. SEIR modeling of the COVID-19 and its dynamics. *Nonlinear Dynamics*, 101(3):1667–1680, June 2020.

Alison J. Heppenstall, Andrew T. Crooks, Linda M. See, and Michael Batty, editors. *Agent-based models of geographical systems*. Springer, Dordrecht, Netherlands, 2012 edition, November 2011.

Herbert W. Hethcote and James Yorke. *Gonorrhea transmission dynamics and control*. Lecture Notes in Biomathematics. Springer, Berlin, Germany, 1984 edition, oct 1984.

Jean-Baptiste Hiriart-Urruty and Claude Lemaréchal. *Fundamentals of Convex Analysis*. Grundlehren Text Editions. Springer-Verlag Berlin Heidelberg, 2001. ISBN 978-3-540-42205-1. doi: 10.1007/978-3-642-56468-0.

M. W. Hirsch and Hal Smith. Chapter 4 monotone dynamical systems. In *Handbook of Differential Equations: Ordinary Differential Equations*, pages 239–357. Elsevier, 2006.

Elizabeth Hunter, Brian Mac Namee, and John Kelleher. A hybrid agent-based and equation based model for the spread of infectious diseases. *Journal of Artificial Societies and Social Simulation*, 23(4), 2020. doi: 10.18564/jasss.4421.

John K. Hunter. Introduction to dynamical systems. Department of Mathematics, University of California at Davis, 2011.

Valeriano Iranzo and Saúl Pérez-González. Epidemiological models and COVID-19: a comparative view. *Hist. Philos. Life Sci.*, 43(3):104, August 2021a.

Valeriano Iranzo and Saúl Pérez-González. Epidemiological models and COVID-19: a comparative view. *Hist. Philos. Life Sci.*, 43(3):104, August 2021b.

Ali Jalali and Sujay Sanghavi. Learning the dependence graph of time series with latent factors. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, page 619–626, Madison, WI, USA, 2012. Omnipress. ISBN 9781450312851.

Hyeontae Jo, Hwijae Son, Hyung Ju Hwang, and Se Young Jung. Analysis of COVID-19 spread in south korea using the SIR model with time-dependent parameters and deep learning. April 2020.

Ioannis Karafyllidis. Design of a dedicated parallel processor for the prediction of forest fire spreading using cellular automata and genetic algorithms. *Eng. Appl. Artif. Intell.*, 17(1):19–36, February 2004.

Ioannis Karafyllidis and Adonios Thanailakis. A model for predicting forest fire spreading using cellular automata. *Ecol. Modell.*, 99(1):87–97, June 1997.

W. O. Kermack and A. G. McKendrick. A contribution to the mathematical theory of epidemics. *Proc. R. Soc. Lond. A Math. Phys. Sci.*, 115(772):700–721, August 1927.

M. Kretzschmar. Measurement and modeling: Infectious disease modeling. In *Reference Module in Biomedical Sciences*. Elsevier, 2016.

Ana Lajmanovich and James A. Yorke. A deterministic model for gonorrhea in a nonhomogeneous population. *Bellman Prize in Mathematical Biosciences*, 28: 221–236, 1976.

Chen-Yu Lee, Patrick W. Gallagher, and Zhuowen Tu. Generalizing pooling functions in convolutional neural networks: Mixed, gated, and tree. In Arthur Gretton and Christian C. Robert, editors, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 464–472, Cadiz, Spain, 09–11 May 2016. PMLR.

Klaus Lehnertz, Timo Bröhl, and Thorsten Rings. The human organism as an integrated interaction network: Recent conceptual and methodological challenges. *Front. Physiol.*, 11:598694, December 2020.

Ka Yin Leung, Frank Ball, David Sirl, and Tom Britton. Individual preventive social distancing during an epidemic may have negative population-level outcomes. *Journal of The Royal Society Interface*, 15(145):20180296, August 2018.

Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–21, 2021. doi: 10.1109/TNNLS.2021.3084827.

Chin-Teng Lin and C. S. George Lee. Neural fuzzy systems: a neuro-fuzzy synergism to intelligent systems. 1996.

Tony Lindeberg. Scale invariant feature transform. *Scholarpedia*, 7(5):10491, 2012. doi: 10.4249/scholarpedia.10491.

Wayne P London and James A Yorke. Recurrent outbreaks of measles, chickenpox and mumps. *Am. J. Epidemiol.*, 98(6):453–468, December 1973.

Douglas A. Luke. Random network models. In *A User's Guide to Network Analysis in R*, pages 147–162. Springer International Publishing, 2015.

Sérgio Machado, Anirudh Sridhar, Paulo Gil, Jorge Henriques, José M. F. Moura, and Augusto Santos. Recovering the graph underlying networked dynamical systems: a deep learning approach. *ArXiv:2208.04405*, 2022. URL `https://arxiv.org/abs/2208.04405`.

Atalanti A. Mastakouri, Bernhard Schölkopf, and Dominik Janzing. Necessary and sufficient conditions for causal feature selection in time series with latent common causes. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 7502–7511. PMLR, 18–24 Jul 2021.

Gonzalo Mateos, Santiago Segarra, Antonio G. Marques, and Alejandro Ribeiro. Connecting the dots: Identifying network structure via graph signal processing. *IEEE Signal Processing Magazine*, 36(3):16–43, 2019. doi: 10.1109/MSP.2018.2890143.

Donatello Materassi and Murti V. Salapaka. Network reconstruction of dynamical polytrees with unobserved nodes. In *Proc. IEEE Conference on Decision and Control (CDC)*, pages 4629–4634, Maui, Hawaii, Dec 2012a. doi: 10.1109/CDC.2012.6426335.

Donatello Materassi and Murti V. Salapaka. On the problem of reconstructing an unknown topology via locality properties of the Wiener filter. *IEEE Transactions on Automatic Control*, 57(7):1765–1777, July 2012b.

Donatello Materassi and Murti V. Salapaka. Identification of network components in presence of unobserved nodes. In *Proc. IEEE Conference on Decision and Control (CDC)*, pages 1563–1568, Osaka, Japan, Dec 2015. doi: 10.1109/CDC.2015.7402433.

Joaquim P. Mateus, Paulo Rebelo, Silvério Rosa, César M. Silva, and Delfim F. M. Torres. Optimal control of non-autonomous seirs models with vaccination and treatment. *Discrete and Continuous Dynamical Systems - S*, 11(6):1179–1199, 2018. doi: 10.3934/dcdss.2018067.

Vincenzo Matta, Augusto Santos, and Ali H. Sayed. Graph learning under partial observability. *Proceedings of the IEEE*, 108:2049 – 2066, 11 2020. doi: 10.1109/JPROC.2020.3013432.

Vincenzo Matta, Augusto Santos, and Ali H. Sayed. Graph learning over partially observed diffusion networks: Role of degree concentration. *IEEE Open Journal of Signal Processing*, pages 1–34, 2022. doi: 10.1109/OJSP.2022.3189315.

Alexandre Mauroy and Jorge Goncalves. Linear identification of nonlinear systems: A lifting technique based on the koopman operator. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 6500–6505, 2016. doi: 10.1109/CDC.2016.7799269.

T. McKelvey, Muhammad Ahmad, Ankur Teredesai, and Carly Eckert. Interpretable machine learning in healthcare. 08 2018.

Jonathan Mei and Jose M. F. Moura. Signal processing on graphs: Causal modeling of unstructured data. *IEEE Transactions on Signal Processing*, 65(8):2077–2092, April 2017. doi: 10.1109/TSP.2016.2634543.

Jonathan Mei and José M. F. Moura. Silvar: Single index latent variable models. *IEEE Transactions on Signal Processing*, 66(11):2790–2803, 2018. doi: 10.1109/TSP.2018.2818075.

Andrea Montanari and José Pereira. Which graphical models are difficult to learn? In *Advances in Neural Information Processing Systems*, volume 22, Vancouver, Canada, 2009.

Ingemar Nåsell. Stochastic models of some endemic infections. *Mathematical Biosciences*, 179(1):1–19, July 2002.

Yutaka Okabe and Akira Shudo. Microscopic numerical simulations of epidemic models on networks. *Mathematics*, 9(9):932, April 2021.

Javier Oltra, Anna Campabadal, Barbara Segura, Carme Uribe, Maria Jose Marti, Yaroslau Compta, Francesc Valldeoriola, Nuria Bargallo, Alex Iranzo, and Carme Junque. Disrupted functional connectivity in PD with probable RBD and its cognitive correlates. *Sci. Rep.*, 11(1):24351, December 2021.

Edward Ott. *Chaos in Dynamical Systems*. Cambridge University Press, August 2002. doi: 10.1017/cbo9780511803260.

Y.-S. Park and S. Lek. Chapter 7 - artificial neural networks: Multilayer perceptron for ecological modeling. In *Ecological Model Types*, volume 28 of *Developments in Environmental Modelling*, pages 123–140. Elsevier, 2016. doi: https://doi.org/10.1016/B978-0-444-63623-2.00007-4.

Padmavathi Patlolla, Vandana Gunupudi, Armin R. Mikler, and Roy T. Jacob. Agent-based simulation tools in computational epidemiology. In *Innovative Internet Community Systems*, pages 212–223. Springer Berlin Heidelberg, 2006. doi: 10.1007/11553762_21.

José Pereira, Morteza Ibrahimi, and Andrea Montanari. Learning networks of stochastic differential equations. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc., 2010.

Liliana Perez and Suzana Dragicevic. An agent-based approach for modeling dynamics of contagious disease spread. *International Journal of Health Geographics*, 8(1):50, 2009. doi: 10.1186/1476-072x-8-50.

William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes 3rd Edition*. 09 2007.

Firda Rahmadani and Hyunsoo Lee. Hybrid deep learning-based epidemic prediction framework of COVID-19: South korea case. *Applied Sciences*, 10(23): 8539, November 2020a.

Firda Rahmadani and Hyunsoo Lee. Dynamic model for the epidemiology of diarrhea and simulation considering multiple disease carriers. *International Journal of Environmental Research and Public Health*, 17(16):5692, August 2020b.

Joseph Ramsey, Madelyn Glymour, Ruben Sanchez-Romero, and Clark Glymour. A million variables and more: the fast greedy equivalence search algorithm for learning high-dimensional graphical causal models, with an application to functional magnetic resonance images. *International Journal of Data Science and Analytics*, 3:121–129, 2016.

Xiao-Long Ren, Niels Gleinig, Dirk Helbing, and Nino Antulov-Fantulin. Generalized network dismantling. *Proceedings of the National Academy of Sciences*, 116 (14):6554–6559, 2019. doi: 10.1073/pnas.1806108116.

Ryan A. Rossi and Nesreen K. Ahmed. The network data repository with interactive graph analytics and visualization. In *AAAI*, 2015. URL `https://networkrepository.com`.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, oct 1986. doi: 10.1038/323533a0.

Augusto Santos, Vincenzo Matta, and Ali H. Sayed. Local tomography of large networks under the low-observability regime. *IEEE Transactions on Information Theory*, 66(1):587–613, January 2020a.

Augusto Santos, Vincenzo Matta, and Ali H. Sayed. Local tomography of large networks under the low-observability regime. *IEEE Transactions on Information Theory*, 66:587 – 613, 01 2020b. doi: 10.1109/TIT.2019.2945033.

Lisa Sattenspiel. Infectious diseases in the historical archives: a modeling approach. In *Human Biologists in the Archives*, pages 234–265. Cambridge University Press, December 2002. doi: 10.1017/cbo9780511542534.012.

Hiroki Sayama. Book: Introduction to the Modeling and Analysis of Complex Systems, jun 23 2019.

Ali H. Sayed. Adaptation, Learning, and Optimization over Networks. *Found. Trends Mach. Learn.*, 7(4-5):311–801, 2014. doi: 10.1561/2200000051.

Santiago Segarra, Antonio G. Marques, Gonzalo Mateos, and Alejandro Ribeiro. Network topology inference from spectral templates. *IEEE Transactions on Signal and Information Processing over Networks*, 3(3):467–483, 2017a. doi: 10.1109/TSIPN.2017.2731051.

Santiago Segarra, Michael T. Schaub, and Ali Jadbabaie. Network inference from consensus dynamics. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 3212–3217, Dec 2017b. doi: 10.1109/CDC.2017.8264130.

G. Ch. Sirakoulis, I. Karafyllidis, and A. Thanailakis. A cellular automaton model for the effects of population movement and vaccination on epidemic propagation. *Ecological Modelling*, 133(3):209–223, September 2000. doi: 10.1016/s0304-3800(00)00294-5.

Peter Spirtes and Clark Glymour. An algorithm for fast recovery of sparse causal graphs. *Social Science Computer Review*, 9(1):62–72, 1991. doi: 10.1177/089443939100900106.

Peter Spirtes, Christopher Meek, and Thomas Richardson. Causal inference in the presence of latent variables and selection bias. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, page 499–506, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search*. MIT press, 2nd edition, 2000.

George Stepaniants, Bingni W. Brunton, and J. Nathan Kutz. Inferring causal networks of dynamical systems through transient dynamics and perturbation. *Physical Review E*, 102(4), October 2020.

Steven H. Strogatz. Nonlinear dynamics and chaos: With applications to physics, biology, chemistry and engineering. pages 1–11, Reading, Massachusetts, 1994. Perseus Books.

Sandra Vaz and Delfim F. M. Torres. A discrete-time compartmental epidemiological model for COVID-19 with a case study for portugal. *Axioms*, 10(4):314, November 2021. doi: 10.3390/axioms10040314.

John Von Neumann and Arthur W. Burks. *Theory of Self-reproducing Automata*. Goldstine Printed Materials. University of Illinois Press, 1966.

Lijing Wang, Jiangzhuo Chen, and Madhav Marathe. TDEFSI. *ACM Transactions on Spatial Algorithms and Systems*, 6(3):1–39, May 2020.

Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, June 1998.

Stephen Wolfram. *Wolfram on Cellular Automata*. Addison Wesley, London, England, March 1994.

Lu Zhao and Yan Wan. Identifiability and estimation of partially-observed influence models. *IEEE Control Systems Letters*, pages 1–1, 2022. doi: 10.1109/LCSYS.2022.3184958.

Jin Zhen and Liu Quan-Xing. A cellular automata model of epidemics of a heterogeneous susceptibility. *Chinese Physics*, 15(6):1248–1256, May 2006. doi: 10.1088/1009-1963/15/6/019.

Jinming Zou, Yi Han, and Sung-Sau So. Overview of artificial neural networks. In *Methods in Molecular Biology™*, pages 14–22. Humana Press, 2008.