Università di Genova

DIBRIS DIPARTIMENTO
DI INFORMATICA, BIOINGEGNERIA,
ROBOTICA E INGEGNERIA DEI SISTEMI

Generalised temporal network inference

Veronica Tozzo

Theses Series DIBRIS-TH-2019-111

Università di **Genova**

Dipartimento di Informatica, Bioingegneria,
Robotica ed Ingegneria dei Sistemi

Ph.D. Thesis in
Computer Science and Systems Engineering
Computer Science Curriculum

Generalised temporal network inference

by

Veronica Tozzo

December, 2019

Ph.D. Thesis in Computer Science and Systems Engineering (S.S.D. INF/01)
Dipartimento di Informatica, Bioingegneria,
Robotica ed Ingegneria dei Sistemi
Università di Genova

Candidate
Veronica Tozzo
veronica.tozzo@dibris.unige.it

Title
Generalised temporal network inference

Advisors
Annalisa Barla
DIBRIS, Università di Genova
annalisa.barla@unige.it

External Reviewers
Carl Henrik Ek
University of Bristol, UK
carlhenrik.ek@bristol.ac.uk

Maurizio Filippone
EUROCOM, SophiaTech
maurizio.filippone@eurecom.fr

Location
DIBRIS, Universitá di Genova
Via Dodecaneso, 35
I-16146 Genova, Italy

Submitted On
December 2019

To Vanessa.
To all the laughters and all the tears
we had together in these three years.

Abstract

Network inference is becoming increasingly central in the analysis of complex phenomena as it allows to obtain understandable models of entities interactions. Among the many possible graphical models, Markov Random Fields are widely used as they are strictly connected to a probability distribution assumption that allow to model a variety of different data. The inference of such models can be guided by two priors: sparsity and non-stationarity. In other words, only few connections are necessary to explain the phenomenon under observation and, as the phenomenon evolves, the underlying connections that explain it may change accordingly.
This thesis contains two general methods for the inference of temporal graphical models that deeply rely on the concept of temporal consistency, *i.e.*, the underlying structure of the system is similar (*i.e.*, consistent) in time points that model the same behaviour (*i.e.*, are dependent). The first contribution is a model that allows to be flexible in terms of probability assumption, temporal consistency, and dependency. The second contribution studies the previously introduces model in the presence of Gaussian partially un-observed data. Indeed, it is necessary to explicitly tackle the presence of un-observed data in order to avoid introducing misrepresentations in the inferred graphical model. All extensions are coupled with fast and non-trivial minimisation algorithms that are extensively validate on synthetic and real-world data. Such algorithms and experiments are implemented in a large and well-designed Python library that comprehends many tools for the modelling of multivariate data.
Lastly, all the presented models have many hyper-parameters that need to be tuned on data. On this regard, we analyse different model selection strategies showing that a stability-based approach performs best in presence of multi-networks and multiple hyper-parameters.

Publications

Some ideas and figures have appeared previously in the following publications.

JOURNAL PUBLICATIONS

Veronica Tozzo, Clohé-Agathe Azencott, Samuele Fiorini and Annalisa Barla. Where do we stand in regularization for life science studies? *Submitted*. (2019)

CONFERENCE PROCEEDINGS

Veronica Tozzo, and Annalisa Barla. "Multi-parameters Model Selection for Network Inference." *International Conference on Complex Networks and Their Applications.* Springer, Cham. (2019)

Federico Tomasi, Veronica Tozzo, and Annalisa Barla. Temporal Patterns Detection in Time-Varying Graphical Models. *Submitted.* (2019)

Federico Tomasi, Veronica Tozzo, Alessandro Verri and Saverio Salzo. Forward-Backward Splitting for Time-Varying Graphical Models. *Proceedings of the Ninth International Conference on Probabilistic Graphical Models, in PMLR* (2018) pp. 72:475-486

Federico Tomasi*, Veronica Tozzo*, Saverio Salzo and Alessandro Verri. Latent variable time-varying network inference. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (2018). pp.2338-2346

Veronica Tozzo, Federico Tomasi, Margherita Squillario and Annalisa Barla. Group induced graphical lasso allows for discovery of molecular pathway-pathway interactions. *Machine Learning for Health (ML4H) Workshop at NeurIPS* (2018) - `arXiv181109673T`

POSTERS AND ORAL PRESENTATIONS

Veronica Tozzo and Annalisa Barla. Modelling of gene expression time series with the latent time-evolving graphical lasso. *3rd Annual MAQC Society Conference* (2019).

Veronica Tozzo, Federico Tomasi, Margherita Squillario, Saverio Salzo and Annalisa Barla. Regularized extension of the latent graphical lasso allows prior imposition. *International seminar and workshop on Stochastic dynamics on large networks: Prediction and inference* (2018).

Acknowledgements

I am extremely grateful to all the people that helped me during my PhD. I would like to thank my advisor, Annalisa Barla and all the members of the research group, who thought me almost everything I know. Special thanks to Federico Tomasi for contributing to the main core of the thesis in many ways through endless discussions, coding tricks and paper writing (his contributions are partially present in Chapter 4, 5, 6 and 7). To Vanessa D'Amario for constant support and collaboration in the analysis on neural data. To Davide Garbarino for all the encouragement in this last month, the theoretical explanations and corrections of both Chapters 4 and 5. To Samuele Fiorini for helping me understand the machine learning basics always and again. To Margherita Squillario for the biological validation on Neuroblastoma data (Chapter 7). I am grateful to Saverio Salzo for teaching me everything I know on optimisation and for the active contribution on Chapter 4, Appendix A and B.
Thanks to all the 309 guys, especially to DOCS. It has been an adventure, you all made it great.
Thanks to my parents that support me, and this research career, even if it will bring me far from home. Thanks to my sister, Laura, for being enthusiast and proud of everything I do, even when I am not. Finally, thanks to Federico for being by my side during these three years with patient, understanding and love.

Contents

List of Figures

List of Tables

List of Algorithms

Introduction

The understanding of complex phenomena is a problem that arises in many applicative fields, such as finance, social science, medicine and biology (Farasat et al., 2015; Hecker et al., 2009; Huang, Liao and Wu, 2016; Liu, Han and Zhang, 2012; Orchard, Agakov and Storkey, 2013). Examples of complex phenomena are the evolution of a disease, the self regulation of the financial market, or, the change in social response to political decisions. All these phenomena can be looked at as systems composed of smaller entities that may or may not act independently.
Often, the study of complex systems is fragmented in simpler tasks. One could look for the set of meaningful entities that are responsible for a specific state of the system (*e.g.*, identifying the genes responsible for the development of a specific cancer), or, one could learn how to predict the behaviour of the system given the observations of its entities (*e.g.*, given a set of gene expressions predict the disease subtype affecting the patient). Nonetheless, these tasks are typically guided by prior knowledge and provide a simple and interpretable solution that, in turn, explains only a small portion of the phenomenon. Therefore, we could say that we are not understanding the system as a whole but we are simply describing some specific aspects of it.
On the contrary, fully understanding a phenomenon entails a comprehension of the dynamic of interactions among entities as well as how these interactions relate to different statuses. Hence, in presence of an explicit relationship between interacting entities and status, the understanding of the phenomenon can be simplified to the observing and learning over time how the entities that are part of the system contribute to a certain effect by interacting with each other.
In this thesis we propose generalised temporal inference methods for the detection of the underlying evolving interactions. Our methods are effective even when the solution computation is compromised by a great number of observed entities.

Context

The most suitable mathematical model for the abstract representation of entities and their interactions is a *graph* or *network*, that provides a compact representation of entities as nodes and their connections as edges. In the ideal case the graph model is known *a priori*, but often it needs to be inferred. The inference can be performed with a variety of different approaches. In this thesis we put ourselves in a machine learning setting: we *observe* the behaviour of (possibly a subset of) the entities within the system and we *infer* the best ap-

proximation of their connections under the form of a graph (Friedman, Hastie and Tibshirani, 2008; Lauritzen, 1996). Such approach is known as *network inference* or *graphical model selection*.
Network inference can be performed through different strategies, typically based on different theoretical assumptions on the meaning of the edges. Here, we consider Markov Random Fields (MRFs) a set of statistical models that consider the connections between entities to describe conditional dependence. To this aim, entities are modelled by random variables that are assumed to follow a proper joint probability distribution.
The inference of a MRF on D variables becomes challenging when we are dealing with large scale data sets, *i.e.*, we have thousands of variables in play. Indeed, it consists in a combinatorial problem of identifying the correct network structure in a search space of possibly $2^{D(D-1)/2}$ edges. Thus, a reliable inference requires a large number of *samples* that are D-dimensional vectors of observations. Nonetheless, typically, the required sample size is not available, therefore the number of variables is much higher than the number of samples ($N \ll D$).
The leading strategy to cope with this issue is to assume that just a reduced number of interactions are actually meaningful to the phenomenon under study and, therefore, constrain the problem and reduce the search space. In particular, we exploit regularised methods that impose a *sparse* prior on the problem (Friedman, Hastie and Tibshirani, 2008; Meinshausen and Bühlmann, 2006). The sparse assumption eases the computational burden allowing us to find an approximate solution. At the same time, given the restricted set of resulting edges, it also improves interpretability of the graph.
While being fundamental for identifiability, regularisation can be also leveraged to extend graphical models in order to consider more complex scenarios as multiple classes, longitudinal data, multi-level networks, latent variables and many other possible conditions (Chandrasekaran, Parrilo and Willsky, 2010; Cheng, Shan and Kim, 2017; Danaher, Wang and Witten, 2014; Geng et al., 2018; Guo et al., 2011; Hallac et al., 2017a). Throughout this thesis we will handle only methods for the inference of MRFs based on a sparse prior that recur to further regularisation strategies to cope with complex settings.

Motivation

Regularised methods for the inference of complex MRFs have been proposed in the last few years in the context of continuous data (*i.e.*, the variables are assumed jointly Gaussian). In literature the so-called Gaussian Graphical Models (GGMs) have been considered in the presence of multi-class data (Danaher, Wang and Witten, 2014; Guo et al., 2011), temporal data (Geng et al., 2018; Hallac et al., 2017a), multi-level networks (Cheng, Shan and Kim, 2017), latent variables (Chandrasekaran, Parrilo and Willsky, 2010) and many others. Here, we mainly focus on temporal graphical models inferred from multi-variate

time-series under different settings as they allow to study an evolving system by modelling the underlying changes of the entities connections.
We argue that considering the temporal component is fundamental in order to being truly able to understand a system. Indeed, as a system evolves the interactions among the variables of which is composed may change as well. Therefore, inferring its underlying structure in a unique steady state could be limiting for the detection of variability patterns. Note that, in reality there are systems for which the most suitable model is a unique structure that remains stable over time. Nonetheless, here, we want to focus on non-stationary systems whose understanding is bound to the observation of their evolution. This is particularly evident in some applications, such as biology, where the interest could be to understand the response of the system to perturbation (Molinelli et al., 2013).
To this aim, Hallac et al., 2017a proposed a regularised extension of a method for the inference of stationary GGMs, the Graphical Lasso (GL) (Friedman, Hastie and Tibshirani, 2008). Such extension allows for the inference of networks at discrete time points connected through a specific dynamical behaviour. This method assumes Markovianity, *i.e.*, each time point is dependent on the previous one. To force such dependency, it employs a temporal consistency function that yields network structures close in time to be similar. This method was shown to improve inference with respect to static methods as it allows to consider the global evolution of the system thus providing a more sound and stable inference of the underlying network. Moreover, it allows to study evolving patterns that are impossible to detect otherwise.
Nonetheless, while being extremely powerful, we point out three aspects as drawbacks of such model:

- it only considers the specific setting of continuous data (Gaussian distribution assumption);
- it only allows for Markovian temporal dependency;
- it does not consider the presence of missing data (Little and Rubin, 2019) which influence how the observable entities are perceived and, hence, which interactions are learned (Choi, Chandrasekaran and Willsky, 2009).

The first is a very common assumption in graphical models as the Gaussian distribution allows for the computation of the joint likelihood on the variables and, thus, an easy inference of the adjacency matrix of the graph Wainwright and Jordan, 2008. Nonetheless, even though it simplifies the inference process it does not allow to model other types of real-world data as count data or binary. In literature, many methods that infer MRFs assuming other distributions exist (Allen and Liu, 2013; Jalali et al., 2011; Ravikumar, Wainwright and Lafferty, 2010; Yang et al., 2012, 2013, 2015), but these often lack of temporal extensions. The second aspect assumes a specific temporal dependency which prevents the user to exploit the knowledge of complex temporal dependency like seasonality. Lastly, the third aspect may perturb the final results and thus, entails the need of inserting missing data assumptions in the inference process to avoid misrepresentations (Meng, Eriksson and Hero, 2014).

Contribution

We propose two major contributions in this thesis that are a step in the direction of filling th aforementioned gaps:

- **Generalised methods for temporal network inference — Chapter 4**. We provide a general statistical model for the inference of graphs that is flexible to diverse distributions, consistency types and possible non-Markovian dependencies. Such method infers a non-stationary graphical model from multi-variate time-series that may have different nature (categorical, binary, counts, continuous) under complex temporal dependency patterns. When we know such patterns *a priori*, we rely on kernels to impose them during inference. When we do not know them, we provide automatic identification techniques.

  Also, we propose a specific instantiation of an *a priori* kernel that allows to transform the problem from temporal network inference to multi-class network inference. Thus, we generalise the multi-class approach for GGMs (Danaher, Wang and Witten, 2014), to other types of distributions beyond Gaussian.

- **Temporal graphical lasso with missing data — Chapter 5**. We propose possible extensions of the temporal network inference with Gaussian assumption in the case of missing data which may either present missing random values or variables that are consistently never measured, and that we define as latent. To solve these two problems we devised two different strategies, one builds on the Expectation Maximisation method and the second on the marginalisation of the latent variables effect. Finally, we show a case in which by exploiting partial prior knowledge on the latent variables, we can obtain results that go in the direction of multi-level networks.

Both our main contributions can be seen as generalisations of state-of-the-art methods that introduce more flexibility and gain in expressivity. Indeed, with these methods we are able to model a wider set of of multi-variate temporal data and to analyse their temporal patterns. During the thesis we provide a thorough assessment of the proposed statistical methods on synthetic data to determine their reliability. Additionally, we show some real-world applications to provide examples of how the proposed methods can be exploited to understand complex dynamics. It is worth mentioning that each method comes with a related optimisation method whose derivation is not a negligible portion of the thesis work. Nonetheless, we defer some of the minimisation algorithms to the Appendices to improve readability.
During the development of our two major contributions we noticed the need to a reliable hyper-parameters selection method. Indeed, all the presented models have more than one hyper-parameter to tune and, given the unsupervised nature of the problem, it may be difficult to detect the best model for a specific data set. Therefore, as a minor contribution we provide a thorough analysis of

the available model selection methods for multiple hyper-parameter selection providing an extension of a stability-based approach (Chapter 3).
Lastly, all the developed code related to this thesis has been developed within a Python framework called *REGAIN* (REgularised GrAph INference) that contains also other inference methods, model selection algorithms, results assessment and plotting utils. We point this out as second minor contribution of this thesis (Chapter 6).

Outline

This thesis comprises four main parts. Part I contains the background on graphical model selection under different probability distribution assumptions (Chapter 1) and the state of the art on graphical models that assume the presence of missing data (Chapter 2). This part is fundamental for the development of the our contributions that are presented in Part II. In Chapter 3, we present model assessment strategies and hyper-parameters selection methods that are widely used throughout the rest of the thesis. Next, we present generalised temporal network inference methods under the assumption of continuous, binary or counts data (Chapter 4). In Chapter 5, we present the methods for the inference of temporal graphical lasso with possibly missing data. In Chapter 6 we present the multi-purpose Python library Regain. In Part III we conclude our thesis by presenting some real-world application examples (Chapter 7) and the conclusions (Chapter 8). In Part IV we provide some additional mathematical details on the more complex optimisation algorithms (Appendix A and B) and the synthetic data generation procedures (Appendix C).

Notation

Unless explicitly specified, we denote with bold lower-case letters $\boldsymbol{x}$ uni-dimensional vectors, with upper case letters X 2-dimensional matrices and with bold upper-case letters $\boldsymbol{X}$ we denote tensors.
The entries of vectors, matrix or tensors are denoted by $x[i]$, $X[ij]$, $\boldsymbol{X}[kij]$ respectively. In the case of tensors we may equivalently write $X_k[ij]$. When we want to select an entire dimension we put a colon, *e.g.*, if we have a 2-dimensional matrix and we want to take the i-th row we write $X[i,:]$. If we want to select all but one row we will write $X[-i,:]$ Given a set of indices $\mathbb{I}_{\mathrm{A}} = \{1, \ldots, j\}$ we denote the squared sub-matrix obtained by selecting the corresponding rows and columns as $X[\mathrm{A}]$. Similarly we denote the sub-matrix obtained by selecting the rows in the set $\mathbb{I}_{\mathrm{A}}$ and the columns in the set $\mathbb{I}_{\mathrm{B}}$ by $X[\mathrm{AB}]$. We will denote the cardinality of the set $\mathbb{I}_{\mathrm{B}}$ with $|B|$.
With $\mathcal{S}^D_{++}$ we denote the cone of positive definite matrices, similarly $\mathcal{S}^D_{+}$ denotes the cone of positive semi-definite matrices. It is equivalent to say that $X \in \mathcal{S}^D_{++}$ or $X \succ 0$, similarly $X \in \mathcal{S}^D_{+}$ is equal to $X \succcurlyeq 0$.
With $\langle \cdot, \cdot \rangle$ we denote the scalar product between two vectors.

PART I

Background

This part contains a description of the context in which this thesis is posed. Chapter 1 presents the steady-state regularised inference methods under different distribution assumptions. Chapter 2 focuses on the concept of missing data, both at random or latent, and the related mechanisms to cope with this problem in steady-state inference methods.

1

Regularized Markov Models

Throughout the thesis we will revolve around the concept of graphs or networks. Nonetheless, a graph could be interpreted differently depending on the meaning we assign on the edges. In particular, Markov Random Fields (MRFs) model conditional probability dependencies between variables. Consider the following example: given two genes A and B they are linked if, given the profiles of all other genes across all the subjects, the levels of genes A are still predictive for the gene B and vice versa. Therefore, a connection in a MRFs has a stronger meaning than correlation. MRFs are widely used in many applications as a mathematical abstraction of a system that allows to straightforwardly study its properties.
Among the methods for the inference of MRFs from data we restrict our focus on those based on the assumption of sparsity, *i.e.*, in high-dimensional contexts the connections that explain the state of the system are few with respect to the total number of possible edges between the variables.
Guided by this prior, the inference of the graph depends on the probability distribution assumed on the data. In this thesis we will analyse three probability distribution: Gaussian, Bernoulli and Poisson that allow to model respectively continuous, binary and counts variables.

OUTLINE In this chapter we briefly introduce the concept of Markov Random Fields (Section 1.1). In Section 1.2 we explain how generalized linear models can be used to infer MRFs starting from those exponential family distributions that have linear sufficient statistics. In Section 1.3 we introduce the problem of penalised network inference. In Sections 1.4, 1.5 and 1.6 we present three distribution-based graphical models (respectively the Gaussian, Ising and Poisson) and in Section 1.7 their state-of-the-art extensions that consider time. We conclude in Section 1.8 with a brief summary of the chapter.

1.1 Markov Random Fields

MRFs are a set of models that belong to the wider set of *probabilistic graphical model*, which, beyond MRF, includes *Bayesian Networks*, *factor graphs* and

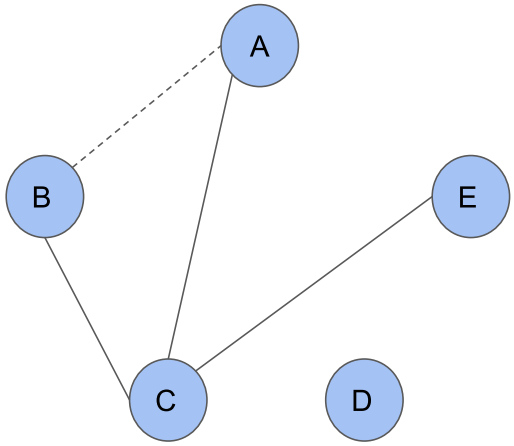


FIGURE 1. *Example of a graphical model where the node A is independent from the nodes B, E, and D given the other nodes. The dashed edge represents the edge we condition away by considering the note C.*

chain graphs (Clifford, 1990; Frey, 2002; Lauritzen, 1996; Murphy and Russell, 2002). Probabilistic graphical models allows a graph to express the conditional dependence structure between random variables, *i.e.*, they define a joint probability distribution on a set of variables. In order to capture the meaning of this set of models we need to first define a graph.

Definition 1 (Graph)**.** A graph is a couple $\mathcal{G} = (V, E)$ where $V = \{1, 2, \ldots, D\}$ is the set of *nodes* or *vertices* and $E = \{(i, j)|i, j \in\}V \times V$ is the set of *edges*. A graph is said *undirected* if there is no distinction between the edge (i, j) and the edge (j, i), otherwise is said *directed*.

The set V of vertices has a bijective correspondence to the set of variables $\{X_1, \ldots, X_D\}$ representing entities of the system in analysis. Then, a MRF is a probabilistic model that factorise according to a graph $G = (V, E)$ in such a way that the conditional dependencies between the variables $X_1, \ldots, X_D$ can be directly read from the edges E. Consider the graph in Figure 1. Here, the nodes A and B are independent given the node C, as no connections between them exists. If the node C was not considered in the inference, we could not condition its presence away and thus, an edge between A and B would appear in the graph (dashed edge).

Definition 2 (Markov Random Field)**.** A Markov Random Field (MRF) is an undirected (possibly cyclic) graph over a set of random variables that satisfies the Markov property.

The Markov properties are:

Definition 3 (Pairwise Markov property)**.** Given two non-adjacent nodes u and v they are conditionally independent given all other variables: $u \perp\!\!\!\perp v|V\backslash\{u, v\}$

Definition 4 (Local Markov property)**.** A variable u is conditionally independent of all other variables given its neighbours denoted by $\mathcal{N}(u)$: $u \perp\!\!\!\perp V\backslash\mathcal{N}(u)|\mathcal{N}(v)$

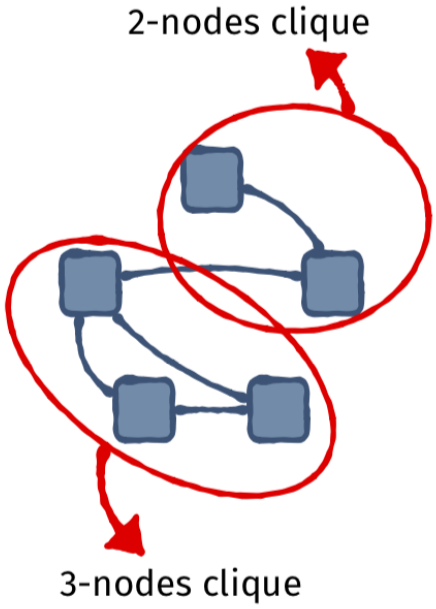


FIGURE 2. *Example of 2-nodes and 3-nodes: small fully connected sub-graphs in an undirected graph of 5 nodes.*

Definition 5 (Global Markov property)**.** Any two sets of variables A and B are conditionally independent given a separating set S that contains all the paths connecting the nodes in A and B: $A \perp\!\!\!\perp B|S$

The properties grow in strength, however they are equivalent in the case of a positive probability (Lauritzen, 1996).

1.1.1 *Gibbs Random Fields*

Definition 6 (Clique)**.** A clique C is a subset of vertices $\bar{V}$ of a graph G such that all the possible couples of nodes in the subset are adjacent, *i.e.*, the corresponding sub-graph $G_{\bar{V}}$ is complete (see Figure 2 for a visual representation). A clique is *maximal* if it is not properly contained within any other clique.

The graph G completely defines a probability distribution p_G over the variables $X_1, \ldots, X_D$. In order to see this more clearly we recall the Hammersley-Clifford theorem (Clifford, 1990), that states that when the probability p_G is strictly positive, a MRF is equal to a Gibbs random field. Therefore, it can be represented as a sum of functions on the graphs cliques.
Given the graph G we can represent the related joint distribution p_G over the variables $X_1, \ldots, X_D$ as the product of compatibility functions that depend only on the subset of variables corresponding to its cliques. Let $\mathcal{C}$ be the set of cliques of the graph G and let $\{\phi_c(X_c), c \in \mathcal{C}\}$ be a set of clique-wise sufficient statistics that depends on the probability assumed on the data then, any dis-

tribution within the graphical model family represented by the graph G, takes the form (Clifford, 1990; Wainwright and Jordan, 2008)

$$p_G(X_1, X_2, \ldots, X_D) \propto \exp\{\sum_{c \in \mathcal{C}} \theta_c \phi_c(X_c)\} \tag{1}$$

Then, given the set $\mathcal{C}$ of cliques of the graph, an MRF is a collection of distributions that factorise as

$$p_G(X_1, X_2, \ldots, X_D) = \frac{1}{A} \prod_{C \in \mathcal{C}} \exp(\theta_c \phi_C(X_C))$$

where A is a log-normalisation constant chosen to ensure that the distribution sums up to 1 (Wainwright and Jordan, 2008). Note that the use of all cliques may be a redundant definition but allows for easier computation while not yielding to loss of generality (Wainwright and Jordan, 2008). Often, it may be useful to use the direct factorisation of the joint probability, defined as

$$p_G(X_1, X_2, \ldots, X_D) = \prod_{v \in V} p(X_v | X_{\mathcal{N}(v)})$$

where $\mathcal{N}(v)$ is the set of variables in the neighbourhood of the variable v.

1.2 Markov Random Fields and the Exponential Families

Exponential families can naturally be interpreted as probabilistic graphical models and more specifically Markov Random Fields. This is due to the fact that exponential families are represented as the summation of weighted functions similarly to the form in Equation (1).

1.2.1 *Exponential Families*

Exponential families are a very flexible family of distributions that includes many of the most known distribution such as Bernoulli, Gaussian, Multinomial, Dirichlet, Gamma, Poisson and Beta.

Definition 7 (Exponential family)**.** Given a sample space $\mathcal{X}^D$ on which it is defined a measure ν and random vector $(X_1, X_2, \ldots, X_D) \in \mathcal{X}^D$we define a collection of functions $\phi = (\phi_\alpha : \mathcal{X}^D \to \mathbb{R})_{\alpha \in \mathcal{I}}$ called *sufficient statistics* to which we associate their *exponential parameters* $\theta = (\theta_\alpha)_{\alpha \in \mathcal{I}}$. Then, the associated exponential family is defined as the following parametrised collection of density functions

$$p_\theta(X_1, X_2, \ldots, X_D | \theta) = \exp\{\langle \theta, \phi(X) \rangle + h(X) - A(\theta)\}$$

where $h(X)$ is a function of only the samples and $A(\theta)$ is the *log normalisation function* that ensures the probability to be properly normalised and it is defined as

$$A(\theta) = \log \int_{\mathcal{X}^D} \exp\langle \theta, \phi(X) \rangle \nu(dX).$$

By fixing the sufficient statistics ϕ we identify a particular type of exponential family (*e.g.*, Poisson or Bernoulli). By also fixing the exponential parameters θ we define a specific member of such family (*i.e.*, a specific probability distribution).

1.2.2 *Exponential-family based Graphical Models*

It is possible to reason in terms of MRFs for any exponential family distribution (Wainwright and Jordan, 2008; Yang et al., 2012, 2015). This representation is particularly suited when the sufficient statistics are linear in the variables as it allows to obtain a straightforward inference algorithm.
Suppose we are given a univariate exponential family distribution

$$p(X) = \exp(\langle \theta, \phi(X) \rangle + h(X) - D(\theta)) \tag{2}$$

with log normalisation function $D(\theta)$.
Consider a D-dimensional random vector $X = (X_1, X_2, \ldots, X_D)$ and an undirected graph $G = (V, E)$ over D variables. Suppose now that the distribution on the variable X_v given the rest of the nodes X_{-v} has the form in Equation (2) with sufficient statistics $\{\phi(X_s)\}_{s \in N(v)}$. Then such distribution is a linear combination of k-th order products of univariate functions

$$\begin{aligned} p(X_v|X_{-v}) = \exp\Big\{ & \theta_v \phi(X_v) - h(X_v) + \bar{D}(X_{-v}) + \\ & \phi(X_v) \Big(\sum_{s \in \mathcal{N}(v)} \theta_{vs} \phi(X_s) \\ & + \sum_{s_2, s_3 \in \mathcal{N}(v)} \theta_{v s_2 s_3} \phi(X_2) \phi(X_3) \\ & + \sum_{s_2, \ldots, s_k \in \mathcal{N}(v)} \theta_{v s_2 \ldots s_k} \prod_{j=2}^{k} \phi(X_{s_j}) \Big) \Big\} \end{aligned} \tag{3}$$

where $h(X_v)$ defines the exponential family and $\bar{D}(X_{-v})$ is the log-normalization constant. By the Hammersley-Clifford theorem the related joint distribution is

$$\begin{aligned} p(X_1, X_2, \ldots, X_D|\theta) = \exp\Big\{ & \sum_s \theta_s \phi(X_v) + \sum_{v \in V} \sum_{s \in N(v)} \theta_{vs} \phi(X_s) \\ & + \sum_{v \in V} \sum_{s_2, s_3 \in \mathcal{N}(v)} \theta_{v s_2 s_3} \phi(X_2) \phi(X_3) \\ & + \sum_{v \in V} \sum_{s_2, \ldots, s_k \in \mathcal{N}(v)} \theta_{v s_2 \ldots s_k} \prod_{j=2}^{k} \phi(X_{s_j})) + h(X_v) - A(\theta) \Big\} \end{aligned} \tag{4}$$

Then, under the assumptions:

1. the joint distribution factorise according to a graph G which has clique-factors of size at most k;

2. the node-conditional distribution follows an exponential family;

the conditional and joint distributions are given by (3) and (4) respectively (Besag, 1974; Clifford, 1990; Wainwright and Jordan, 2008; Yang et al., 2012, 2015).
This strictly connects exponential family with MRFs as the exponential parameters are nothing else but the connections of cliques of different size in the graph.

When the joint distribution has factors of size at most two ($k = 2$) and the sufficient statistics are linear functions $\phi(X_v) = X_v$ than the conditional distribution is a generalized linear model Nelder and Wedderburn, 1972 with conditional distribution of the form

$$p(X_v|X_{-v}) = \exp\left\{\theta_v X_v + \sum_{s\in\mathcal{N}(v)} \theta_{vs} X_v X_s + h(X_v) - \bar{D}(X_{-v}, \theta)\right\} \tag{5}$$

and joint distribution

$$p(X_1, \ldots, X_D|\theta) = \exp\left\{\sum_v \theta_v X_v + \sum_{(v,s)\in E} \theta_{vs} X_v X_s + \sum_{X_v} h(X_v) - A(\theta)\right\} \tag{6}$$

1.3 Network Inference

Network inference or *graphical model selection* aims at selecting the most probable graph from observations of the variables. It arises in lots of applications when the underlying graph structure of variables is not known (Barabasi and Oltvai, 2004). Suppose to have D random variables denoted $X_1, \ldots, X_D$ of which we can observe N samples $X \in \mathcal{X}^{N\times D}$ where $X[i,:] = (X[i1], \ldots, X[iD])$ for $i = 1, \ldots, N$. We aim at inferring the set E of edges of the graph $G = (V, E)$, that better fits the data.
Such goal can be reached with inference methods based on Maximum Likelihood Estimation (MLE) . The concept of MLE relies on the maximization of a likelihood function of the model. Indeed, such value tells us how much observations are likely given a model defined by a set of parameters. In our case the model corresponds to the graph G, whose parameters θ we represent by its adjacency matrix K.

Definition 8 (Likelihood)**.** Given data X and the parameters K the *likelihood* $L(X|K)$ of the graph G is any function of K proportional to the density function $p(X|K)$.

Note that the likelihood is a function of the parameter K for fixed X whereas the probability is a function of X for fixed K.
It is common, for optimisation problems, to use the *log-likelihood* $\ell(X|K)$ (the natural logarithm of the likelihood function $L(X|K)$) as it allows to remove products and exponentials within the function to optimise, plus avoiding numerical issues when $D \ll N$.

Suppose now that, for fixed data X, we have two possible values for the parameters K, K' and K'', and that $L(X|K') \geq L(X|K'')$. This means that K' is at least likely as K'' and, therefore, it is the one the better supports the data. This naturally leads to the concept of Maximum Likelihood:

Definition 9 (Maximum Likelihood)**.** A maximum likelihood estimate of K is a value, K^*, that maximizes the likelihood $L(X|K)$ — or the log-likelihood $\ell(X|K)$.

1.3.1 ℓ_1 *Penalisation*

The inference of graphical models through MLE still remains a difficult problem given the dimension of the possible search space. If we consider D variables, it is combinatorial in the number of possible edges, $2^{D(\frac{D-1}{2})}$. We can reduce the search space by assuming *sparsity* of the solution. Such assumption, by constraining the problem, eases the identification of the graph, improves interpretability of the results and reduces the noise. It is fundamental especially when the number of variables is higher than the number of available samples (the so-called $D \gg N$ problem).
Formally, this translates into the addition to a MLE problem of a sparsity-enforcing penalty called ℓ_1-norm. Such norm is a convex non-smooth function that is often used as a relaxation of the non-convex ℓ_0-norm that enforces the number of edges to be small. Given the adjacency matrix K of the graph G, the ℓ_1-norm is defined as

$$\|K\|_{1,od} = \sum_{ij} |K_{ij}| \tag{7}$$

Such norm penalises the weight of edges between the variables shrinking their value and forcing those edges that have values in an interval $[-\alpha, \alpha]$ to be zero, thus selecting only a subset of possible connections. Here, α measures the strength of the penalty on the problem, the higher the α the higher the number of zero edges (Hastie, Tibshirani and Wainwright, 2015).
Such penalisation approach has been widely used in literature, and the very first model exploiting such idea was proposed and developed by (Meinshausen and Bühlmann, 2006) for a neighbourhood estimation of Gaussian Graphical Models.

Definition 10 (Neighbourhood estimation)**.** Penalised neighbourhood estimation infers the conditional independence separately for each node in the graph solving a lasso-like problem where the considered variable is the dependent variable and the others are considered as independent covariates.

Consistency proofs of such approach are provided (Meinshausen and Bühlmann, 2006), and they can be extended for logistic regression (Wainwright, Lafferty and Ravikumar, 2007). In particular, in (Wainwright, Lafferty and Ravikumar, 2007) they provide sufficient conditions on the number of samples, dimensions and neighbourhood size to estimate the neighbourhood of each node

simultaneously. Neighbourhood estimation, nevertheless, has been shown to be an approximation of the exact problem (Friedman, Hastie and Tibshirani, 2008; Yuan and Lin, 2007) as it does not yield to the MLE when there is no equality between the (possibly perturbed) empirical covariance matrix and the estimated one. In (Friedman, Hastie and Tibshirani, 2008) the authors bridge the conceptual gap between this and the exact problem proposing the graphical lasso method, based on the work of (Banerjee, Ghaoui and d'Aspremont, 2008). Later, the concept of an ℓ_1 penalised MLE was proposed also for non-Gaussian distributions (Banerjee, Ghaoui and d'Aspremont, 2008). Again inference is performed via neighbourhood selection as the computation of the joint likelihood is infeasible (Bien and Tibshirani, 2011; Meinshausen and Bühlmann, 2006; Ravikumar, Wainwright and Lafferty, 2010; Ravikumar et al., 2011; Wainwright, Lafferty and Ravikumar, 2007; Yang et al., 2012, 2013; Yuan and Lin, 2007).

OTHER PENALISATION Adding constraints through penalisation also offers the possibility to impose prior information on the resulting graph. All the regularised methods for graph inference assume sparsity. In literature, though, we can find other penalties that allow to force a specific behaviour on networks. In particular, Guo et al., 2011; Honorio and Samaras, 2010; Kolar et al., 2010; Varoquaux et al., 2010; Xie, Liu and Valdar, 2016 estimate multiple networks at one by using a group lasso norm (ℓ_{21}) which helps with the joint selection of features across multiple graphs. Cheng, Shan and Kim, 2017, instead, by imposing a group lasso norm learn a bipartite network of features and group of features. Chandrasekaran, Parrilo and Willsky, 2010 uses a low rank norm to learn a marginalisation of the network that allows to subtract the contribute of latent variables. Hallac et al., 2017a enforce temporal similarity between consecutive graph in time. This latter idea is the main idea to which temporal network inference relies on and will be explained in Section 1.7 and massively used throughout the thesis.

1.3.2 *Penalized MLE for Generalized Graphical Models*

We perform network inference exploiting a penalised version of MLE. In particular, we aim at estimating the parameters of a graphical model based on exponential family distribution that has joint probability specified as in Equation (6). We estimate the graph G via neighbourhood estimation as, except for the Gaussian distribution, we are not able to compute the normalisation factor that allows us to consider the joint distribution of the variables.
Thus, we will reason in terms of the conditional distribution in Equation (5). It can be rewritten considering the adjacency matrix K formed by the parameters of the distribution. In particular we define

$$K[vs] = \begin{cases} \theta_{vs} & \text{if } v \neq s \\ 0 & \text{otherwise} \end{cases} .$$

Then the conditional probability is defined as

$$p_G(X_v|X_{-v}) = \exp\left\{ X_v\Big(\sum_{s\in\mathcal{N}(v)} K[vs]X_s\Big) + h(X_v) - D\Big(\sum_{s\in\mathcal{N}(v)} K[vs]X_s\Big)\right\}$$

and the related conditional log-likelihood on each variables is defined as

$$\begin{aligned}\ell(X|K[v:]) &= -\frac{1}{N}\log\prod_{i=1}^{N} p_G(X[iv]|X[i,-v],K[vs]) \\ &= -\frac{1}{N}\sum_{i=1}^{N} X[iv](\langle K[v:], X^\top[i,:]\rangle) - D(\langle K[v:], X^\top[i,:]\rangle).\end{aligned}$$

Then, the likelihood of the global model is defined as a summation of the likelihoods on the single variables

$$\ell(X|K) = \sum_{v\in V} \ell(X|K[v:])$$

and it adapts to different distributions depending on the function D. The penalised MLE of the adjacency matrix K is then performed by minimising the following functional

$$\underset{K}{\text{minimise}} \quad -\ell(X|K) + \lambda\|k\|_1. \tag{8}$$

Such problem is separable, given the definition of the global likelihood. Therefore, it is possible to estimate the neighbours of each node individually and then merge them together. Each neighbourhood for a variable X_v is obtained by observing the non-zero entries of the row $K[v,:]$. Note that, in this way the neighbourhood selection is not symmetric, *i.e.*, the matrix K is non-symmetric and, therefore, while the variables v and s may be estimated to have an edge when we are minimising on the node v, this may not happen when we minimize on the node s. Thus, the final graph has to be obtained by merging the two neighbourhoods. This can be done either by union or intersection of the edges. Therefore the value of the edge $(v,s) = (s,v)$ to make the matrix K symmetric is determined as

$$\max/\min\{|\text{sign}(K[v,s]|, |\text{sign}(K[s,v])|\} \quad \forall v \neq s \tag{9}$$

where we use the maximum if we want to use the union and the minimum if we want to intersect. Note that it is possible to circumvent the problem by imposing a symmetric constraint and optimising all the variables together rather then separately. We will use this concept in Chapter 4 for the inference of temporal models.

ALTERNATIVE METHODS Often, in literature, we can find methods that exploit Gaussian based inference plus some transformation based on *copula* (Liu, Lafferty and Wasserman, 2009; Liu et al., 2012) or *log2* transforms (Changyong et al., 2014). Such methods leverage on the properties of the Gaussian

distribution that allows to minimise directly the joint distribution in a convex optimisation problem Friedman, Hastie and Tibshirani, 2008. Nonetheless, these methods were proved to be less effective in retrieving the original graph than methods based on more appropriate distributions and, as such, we will not discuss these methods in depth as the probability distributions we analyse are sufficient to model the vast majority of real-world data.

RELATION WITH MAXIMUM A POSTERIORI ESTIMATES Typically, the inference is performed as a Maximum Likelihood Estimate because it simplifies the computation. Nonetheless, such problem is strictly connected to a Maximum A Posteriori (MAP) problem (Murphy, 2012). Indeed, given an assumed probability distribution $p(X|K)$ and a Laplace prior (Kotz, Kozubowski and Podgorski, 2012) of the form

$$\text{Laplace}(\mu, b) = \frac{1}{2b} e^{-\frac{|x-\mu|}{b}}$$

we can write the problem as follows

$$\underset{K}{\text{argmax}} \left[\log \prod_{i=1}^{N} p(X_i|K) + \log \prod_{j=0}^{\Gamma} \frac{1}{2b} e^{-\frac{|K_j|}{b}} \right]$$

where Γ is the set of indices of the upper-triangular of K.
It can be shown that this has the exact same form of Equation (8). Indeed, if we instantiate the probability with the Gaussian we obtain

$$\begin{aligned}
&\underset{K}{\text{argmax}} \left[\log \prod_{i=1}^{N} \frac{1}{(2\pi)^{D/2} \det(K^{-1})^{1/2}} \exp\left(-\frac{1}{2} X_i K X_i^\top \right) + \log \prod_{j=0}^{\Gamma} \frac{1}{2b} e^{-\frac{|K_j|}{b}} \right] = \\
&\underset{K}{\text{argmax}} \left[\sum_{i=1}^{N} -\frac{D}{2} \log(2\pi) + \frac{1}{2} \log \det(K) - \frac{1}{2} X_i K X_i - \frac{1}{b} \sum_{j=0}^{\Gamma} |K_j| \right] = \\
&\underset{K}{\text{argmin}} \left[\text{tr}(X^\top X K) - N \log \det(K) + \frac{2}{b} \sum_{j=0}^{\Gamma} |K_j| \right] = \\
&\underset{K}{\text{argmin}} \left[\text{tr}(X^\top X K) - N \log \det(K) + \lambda \sum_{j=0}^{\Gamma} |K_j| \right]
\end{aligned}$$

Such Bayesian perspective of the problem can also be adopted in the case of joint network inference which is our main setting. A detailed example is provided in Li, McCormick and Clark, 2018, where they present a new class of priors that allows to perform group and fused graphical lasso similarly to the MLE approach proposed in (Danaher, Wang and Witten, 2014).

1.3.3 *Sparsistency and persistence*

To define a model is useful to study its properties in terms of *sparsistency* and *persistence* (Lam and Fan, 2009). The first one was introduced in (Ravikumar

et al., 2009b) and it is the shorthand for "consistency of the sparsity pattern of a parameter", which in our case are the edges E. The second one is defined in (Greenshtein and Ritov, 2004) and it is basically a consistency of the risk which estimates how many edges we need to infer so that the risk of the inferred graph is close to the best graphs in the search space S. These two should be analysed to establish sufficient conditions when the parameters of the model vary with the number of observations N. For parameters we mean the number of nodes D, the maximum node degree D and the size of the search space M. The sparsistency is defined as (Ravikumar et al., 2009b)

$$\mathbb{P}[\hat{E}_N = E^*] \to 1 \quad N \to +\infty$$

while the persistence is defined as (Greenshtein and Ritov, 2004)

$$\mathbb{E}[-\ell(\hat{K}_N|X)] - \inf_{K \in S} \mathbb{E}[-\ell(X|K)] \to 0.$$

In this thesis we will not provide such results on the proposed models as they are still under investigation. We include the definition here for completeness and to help the reader whenever these concepts are mentioned.

1.4 Gaussian Graphical Models (GGMs)

Gaussian Graphical Models (GGMs) are widely used, for example in psychology (Damaraju et al., 2014; Epskamp, Borsboom and Fried, 2018), biology (Jones et al., 2011; Stegle, Teichmann and Marioni, 2015) and neurology (Smith et al., 2011) and they are particularly suited for the modelling of continuous variables. This means that the sample space is defined as $\mathcal{X} = \mathbb{R}$.
GGMs are probabilistic graphical models which variables are jointly distributed according to a multivariate Gaussian distribution $\mathcal{N}(\mu, \Sigma)$ where $\mu \in \mathbb{R}^D$ is the mean vector, and $\Sigma \in \mathcal{S}^D_{++}$ is the $D \times D$ covariance matrix. For simplicity, unless otherwise specified, throughout this thesis the normal distributions are assumed to be centred in 0, *i.e.*, $\mu_i = 0 \ \ \forall i = 1, \dots, D$. This is assumed without loss of generality as we let the variables to be completely explained by the covariance matrix Σ (Choi et al., 2011).
If Σ is a well-defined covariance matrix, (*i.e.*, is positive definite), then the conditional independence between variables in the multivariate normal distribution is associated to zero entries in its inverse (Dempster, 1972).

Proposition 1. Let $X \sim \mathcal{N}(0, \Sigma)$ be a random vector drawn from a multivariate normal distribution, where $K = \Sigma^{-1}$ is the precision matrix of the distribution. Let Γ be the set of entries in Σ. Then, for each $v, s \in \Gamma$ with $v \neq s$,

$$X_v \perp\!\!\!\perp X_s | \Gamma \setminus \{v, s\} \to K[vs] = 0.$$

This result follows from standard linear algebra, details and proof of the proposition can be found in (Lauritzen, 1996, Section 5.1.3). Therefore, the precision matrix is associated to the graph G, where an edge exists if and only if the

two variables have a value different than zero in the corresponding entry of the precision matrix K. For this reason, the precision matrix can be considered as the weighted adjacency matrix of G, encoding the conditional dependences between variables.

The Gaussian distribution is the only exponential family distribution for which it is feasible to compute the normalisation constant. Indeed, such value is defined as the integral over all the possible values of the random variables, and, only with the Gaussian distribution, we can compute a finite form Wainwright and Jordan, 2008. Therefore, this allows to reason in terms of joint distribution which leads to more consistent results (Friedman, Hastie and Tibshirani, 2008). Let $X = (X_1, \dots, X_D) \sim \mathcal{N}(0, \Sigma)$ indicate a random vector, and X the dataset containing N realisations of the D variables in such a way that $X \in \mathbb{R}^{N \times D}$, then the density function is defined as

$$p(X|\Sigma) = \frac{1}{(2\pi)^{D/2}\det(\Sigma)^{1/2}} \exp\left(-\frac{1}{2} X\Sigma^{-1}X^\top\right). \tag{10}$$

Given the set of N *iid* samples in D dimensions in matrix X, based on Equation (10), the Gaussian log-likelihood is defined as

$$\ell(X|\Sigma) = \log \prod_{i=1}^{N} p(X_i|\Sigma) = -\frac{N}{2}\log\det(\Sigma) - \frac{N}{2}\operatorname{tr}\left(\frac{1}{N}X^\top X\Sigma^{-1}\right).$$

This likelihood is expressed in function of Σ but, in literature, it has been shown that estimating the precision matrix K leads to better results (Banerjee, Ghaoui and d'Aspremont, 2008; Bien and Tibshirani, 2011; Friedman, Hastie and Tibshirani, 2008; Lauritzen, 1996; Meinshausen and Bühlmann, 2006; Ravikumar et al., 2011; Wainwright, Lafferty and Ravikumar, 2007; Yuan and Lin, 2007). We re-write the likelihood in terms of the precision matrix K as

$$\ell_{GGM}(X|K) \propto N\log\det(K) - \operatorname{tr}\left(\frac{1}{N}X^\top XK\right) + c \tag{11}$$

where log det denotes the logarithm of the determinant of the matrix K, tr is the trace function defined as $\operatorname{tr}(\cdot) = \sum_i (\cdot)[ii]$, *i.e.*, the sum of the diagonal elements of the matrix and c is a constant term.

1.4.1 *Lasso Penalisation*

Given the sparsity assumption we want to force some entries of the precision matrix K to be zero, as introduced in Section 1.3.1 (Meinshausen and Bühlmann, 2006) . A model for the inference of K including the sparse prior is the *graphical lasso* (GL) what writes out as (Friedman, Hastie and Tibshirani, 2008; Hastie, Tibshirani and Wainwright, 2015):

$$\underset{K}{\text{minimize}} \; -\ell_{GGM}(K|X) + \alpha \left\|K\right\|_{od,1}, \tag{12}$$

where $\|\cdot\|_{od,1}$ is the off-diagonal ℓ_1-norm. Equation (12) has a lasso-like form (Tibshirani, 1996). For this reason, the problem can be solved by coordinate descent, using a modified lasso regression on each variable in turn, thus leading to a simple, efficient and fast procedure.

The graphical lasso has been shown to have good asymptotic properties in terms of persistence and sparsistency (Ravikumar et al., 2011; Rothman et al., 2008).

1.5 Ising Graphical Models (IGMs)

The Ising Graphical Model (IGM) is suited for the modelling of binary or categorical variables (Jalali et al., 2011; Ravikumar, Wainwright and Lafferty, 2010) and it is the most used example of pairwise graphical model as it is derived from Bernoulli variables of which we consider only the pairwise interaction (Bresler, 2015; Ravikumar, Wainwright and Lafferty, 2010). The Bernoulli assumption can be applied on a variety of different applications: voting patterns (Banerjee, Ghaoui and d'Aspremont, 2008), single nucleotide genetic mutations, neural spikes (Schneidman et al., 2006), gases or magnets in statistical physics (Ising, 1925), computer vision (Geman and Geman, 1987) and social network analysis. The Bernoulli distribution belongs to the class of exponential family distribution, therefore this method can be solved via Generalized Linear Model (Yang et al., 2012) (as we introduced in Section 1.2).
An IGM assumes each variable to take values $X \in \{0, 1\}$, and to have density

$$p(X) = p^X (1-p)^{1-X}$$

where p is the probability of x to assume value equal to 1.
Consider now D binary variables sampled in the space $\mathcal{X} = \{0, 1\}^D$ (or equivalently $\{-1, 1\}^D$), it can be easily shown that the sufficient statistic for their univariate Bernoulli distribution is X_i and, if we only consider the pair-wise interactions between them, the conditional probability of a variables X_v is again a Bernoulli with probability

$$p = \sum_{s \in \mathcal{N}(v)} K[vs] X_s$$

that, if we denote x_v as the realisation of X_v, writes out as

$$p_{IGM}(x_v | x_{-v}) = \frac{\exp(c x_v \sum_{s \in \mathcal{N}(v)} K[vs] x_s)}{\exp(c x_v \sum_{s \in \mathcal{N}(v)} K[vs] x_s) + 1} \tag{13}$$

for all $v = 1, \ldots, D$ where $c = 1$ if $X_v \in \{0, 1\}$ and $c = 2$ if $X_v \in \{-1, 1\}$.
Then, the joint IGM distribution has a linear sufficient statistics $\phi(X_s, X_t) = X_s X_t$ and is defined as:

$$p_{IGM}(X|K) = \frac{1}{A(K)} \exp\left\{ \sum_{v \in V} \sum_{s \in \mathcal{N}(v)} K[vs] X_v X_s \right\}, \tag{14}$$

where the normalisation constant $A(K)$ is intractable and cannot be directly computed.

Suppose now that we are given a collection of N samples in the form of a matrix $X \in \{-1,1\}^{N\times D}$ where each row $X_i \in \{-1,1\}^D$ is *iid* and drawn from the distribution p_{IGM} of the form in Equation (14). The conditional log-likelihood of this problem is defined as

$$\begin{aligned}\ell_{IGM}(X|K) &= -\frac{1}{N}\sum_{v\in V}\log p_G(X[:v]|X[:,-v]) \\ &= -\frac{1}{N}\sum_{v\in V}\Bigg\{\Big[\log\Big(\exp(K[v,:]X_{-v} + \exp(-K[v,:]X^\top\Big)\Big] \\ &\qquad - K[v,:]\mu[vs]\Bigg\}\end{aligned} \tag{15}$$

where $\mu[vs] = \frac{1}{N}X[:,v]^\top X[:,s]$ are empirical moments.

1.5.1 *Lasso penalisation*

Given the sparsity assumption we can force some elements of the matrix K to be zero. Note that the conditional distribution in Equation (13) can be interpreted as a logistic regression problem where the response of the variable X_v is the output of a classification task where the other $D-1$ variables are the covariates (Ravikumar, Wainwright and Lafferty, 2010) — similarly to the neighbourhood selection in GGMs (Meinshausen and Bühlmann, 2006). The regularised regression problem then becomes

$$\underset{K\in -1,0,1^{D\times D}}{\text{minimise}} \; -\ell_{IGM}(X|K) + \alpha\|K\|_{1,od} \tag{16}$$

that can be easily minimised for each variables via neighbourhood selection. The final graph is then retrieved unifying the neighbourhoods following the method in Equation (9). Such problem can be solved separately for each variable by solving a logistic regression problem that, given the ι iterations needed for convergence has an inner complexity of N. Thus, the complexity of the problem is $O(DN\iota)$.

1.6 Poisson Graphical Models (PGMS)

Poisson Graphical Models (PGMs) were proposed to satisfy the need of modelling *counts data* properly (Allen and Liu, 2013; Yang et al., 2013). Indeed, with the advances in Next Generation Sequencing (Metzker, 2010) a lot of graphical model based on non-parametric GGMs (Nikoloulopoulos and Karlis, 2009a,b) were used without exploiting the nature of NGS data that born as counts. Nonetheless, other data present the same peculiarity for example climate studies, user-ratings data, term-document counts, site visits and many others. In (Yang et al., 2012, 2015) they proposed a model that exploits the sums of independent Poisson variables which becomes easily intractable with high-dimensional data and it is able to model only positive correlations. Later on in (Yang et

al., 2013) they proposed a Truncated Poisson Distribution to overcome this issue showing promising results. We rely on the model proposed in (Allen and Liu, 2013) where the authors use the exponential family idea to obtain a joint distribution that is based only on the neighbourhood of each node therefore satisfying only the Local Markov Property instead of the global.
Given D variables $X_1, \dots, X_D$ taking values in the space $\mathcal{X} = \mathbb{N}$, whose realisations are denoted as $x_1, \dots, x_D$, each is assumed to have a univariate Poisson distribution parameter λ_v for $v = 1, \dots D$ defined as

$$p(x_v) = e^{-\lambda_v} \frac{\lambda_i^{x_v}}{x_v!}$$

and has sufficient statistics $\phi(x_v) = (x_v, -\log(x_v))$. The joint PGM distribution is

$$p_{PGM}(X|K) = \exp\left\{ \sum_{v \in V} (-\log(X_v!)) + \sum_{v \in V} \sum_{s \in \mathcal{N}(v)} K[vs] X_v X_s - A(K) \right\}$$

where the normalization constant $A(K)$ has to be finite to ensure the probability to be well-defined. Such quantity is defined as (Allen and Liu, 2013)

$$A(K) = \log \left[\sum_{X_v, X_s \in \mathcal{X}} \exp\left(\sum_{v \in V} (-\log(X_v!)) + \sum_{v \in V} \sum_{s \in \mathcal{N}(v)} K[vs] X_v X_s \right) \right].$$

The term $K[vs]X_vX_s$ dominates the summation and, thus, must be finite for infinite values of X_v and X_s. This implies that for $A(K)$ to be finite $K[vs] \leq 0$, $\forall v \neq s$. Therefore, the PGM is only able to detect negative conditional dependencies (Yang et al., 2012, 2015).
This restriction in the joint distribution prevents its direct use, the model proposed in (Allen and Liu, 2013) overcome this issue by basing the model only on the node-conditional Poisson distribution without specifying a joint model. The conditional distribution is defined as

$$p(X_v|X_{-v}, K) = \exp\left\{ -\log(X_v!)) + \sum_{s \in \mathcal{N}(v)} (K[vs] X_v X_s - A(K[vs])) \right\}$$

here $A(K[vs])$ is the log-partition term of the Poisson distribution, computed as the union of the local conditional Poisson modules for each variable:

$$\begin{aligned} A(K[vs]) &\approx \log[\mathbb{E}(X_v|X_s = x, \forall s \in V \backslash v, K)] \\ &= \sum_{s \in \mathcal{N}(v)} (K[vs] X_v X_s) \end{aligned}$$

which satisfies both the pair-wise and local Markov properties (Allen and Liu, 2013).

As before, given N observations of D variables in the form of a matrix $X \in \mathbb{N}^{N\times D}$ the conditional log-likelihood is given by

$$\begin{aligned}\ell_{PGM}(X|K) = &-\frac{1}{N}\sum_{v\in V}\sum_{i=1}^{N}\log\, p(X[i,v]|X[i,-v]) = \\ &-\frac{1}{N}\sum_{v\in V}\sum_{i=1}^{N}\Big[X[iv]X[i,:]K^{\top}[v,:] - \exp(X[i,:]K^{\top}[v,:])\Big]\end{aligned} \tag{17}$$

1.6.1 *Lasso penalisation*

The inference problem can be solved through penalised MLE as previously done for the Gaussian and the Ising models (Banerjee, Ghaoui and d'Aspremont, 2008; Ravikumar, Wainwright and Lafferty, 2010). The minimisation problems writes out as

$$\underset{K\in\{0,1\}^{D\times D}}{\text{minimise}} \; -\ell_{PGM}(X|K) + \alpha\|K\|_{1,od}, \tag{18}$$

and can be solved through neighbourhood estimation followed by the intersection or union of the results of the inferred neighbourhood (Equation (9)).
The optimisation algorithm has, similarly to the Ising model, a complexity of $O(DN\iota)$ where ι are the iterations needed for convergence.

1.7 Temporal extensions

Problems in Equation (12), (16) and (18) aim at recovering the structure of the system at fixed time (*static network inference*). However, complex systems may have temporal dynamics that regulate their overall functioning (Albert, 2007). Hence, the modelling of such complex systems requires a *dynamical network inference*, where the states of the network are intended as co-dependent.
Indeed, the analysis of a set of variables which describe the system at a particular time point could not provide enough information on the more global and general behaviour of the system. As an example, one may consider the analysis of genes observations under the presence of a particular phenotype. Static network inference would answer to the question regarding a particular status of the cell. The answer to the same question asked later in time could lead to a different answer.
The idea of time-varying network inference is to continue the inference process in time. It could be seen as a generalisation of a static inference process that infers separately networks at different point in time. The addition is that time-varying network inference exploits the temporal component during the optimisation. This can improve performances as, in static network inference, there is no theoretical guarantee that the network at step t would be similar to the network at step $t+1$, while one may intuitively expect so. Dynamic network inference instead will embed prior knowledge on the evolution of the network which could help in presence of noise in particular time points of the

network. Indeed, changes in the network at a particular time point may be due to external perturbation, noise or a particular developing state of the system. The dynamism can be modelled in different ways:

1. by assuming a specific temporal dynamic modelled by differential equations (Abegaz and Wit, 2013; Hertz, Roudi and Tyrcha, 2011) ;
2. by assuming stochasticity on the edge of the networks (Geng et al., 2018; Pereira, Ibrahimi and Montanari, 2010);
3. by assuming a temporal consistency modelled through a similarity function between contiguous time points (Bianco-Martinez et al., 2016; Hallac et al., 2017a).

The first and second options are suitable for many applications but they, in turn, require a wide knowledge on the applicative domain. Temporal consistency, instead, allows us to be broad on the possible applications. In this thesis, as we do not have a specific domain in mind, we focus and exploit the concept of temporal consistency.

1.7.1 *Temporal consistency*

In order to exploit temporal consistency we need to assume that the network models a non-stationary distribution that may change at each time point. This implies that to different time points correspond different states of the system that cannot be expressed by a unique model.
We assume a consistency (or similarity) between consecutive states of the network, as, for sufficiently close time points, a system would show negligible differences (Hallac et al., 2017a). The same consistency principle can be exploit in other contexts as multi-class (Danaher, Wang and Witten, 2014; Guo et al., 2011) or multi-layer (Cheng, Shan and Kim, 2017) network inference.

Definition 11 (Consistency)**.** Two inferred networks are said to be exactly consistent if the distance between the related network structures, in terms of some norm, is zero.

The more the distance grows the less consistent the networks are.
The inference of a dynamical network that assumes temporal consistency of consecutive time points can be performed through a regularised approach that extends the stationary model with the imposition of a penalty (Gibberd and Roy, 2017).
The main example of this type of dynamical inference is the context of GGMs and it is the *time-varying graphical lasso* (TVGL) (Hallac et al., 2017a) where the inference of a network at a single time point t is guided by the states at adjacent time points. Throughout the thesis this model will be also referred to as TGL. When we mention it as TVGL we intend the original version proposed in (Hallac et al., 2017a) while when we call it TGL we intend our re-implementation of such model. We will try to explicitly mention this when there is an ambiguity.

Consider now a system formed by D entities measured over T time points. For each time point t we have N_t samples randomly drawn as

$$\boldsymbol{X} = \big(X_1, \ldots, X_T\big) \sim \big(\mathcal{N}(0, \Sigma_1), \ldots, \mathcal{N}(0, \Sigma_T)\big)$$

where $X_t \in \mathbb{R}^{N_t \times D}$ for $t = 1, \ldots, T$.
The goal, since we are in the Gaussian case, is to infer the *precision matrices* $\boldsymbol{K} = \big(K_1, \ldots, K_T\big) \in \mathbb{R}^{\big(D \times D\big) \times T}$. that encode the conditional dependencies at each time point (Hallac et al., 2017a). The TVGL problem is defined as follows:

$$\underset{K_t \in \mathcal{S}^D_{++}}{\text{minimize}} \quad \sum_{t=1}^{T} -N_t \; \ell_{GGM}(X_t | K_t) + \alpha \, \|K_t\|_{\text{od},1} + \beta \sum_{t=1}^{T-1} \Psi(K_{t+1} - K_t), \quad (19)$$

where Ψ is a function that encodes prior information on the temporal behaviour of the network. The related parameter β imposes a certain strength on the consistency of such behaviour in time.

PENALTY FUNCTIONS Hallac et al., 2017a proposed different consistency functions that guarantee a fast optimisation of the related problem. In particular, we can choose among:

- $\Psi(\cdot) = \ell_1(\cdot) = \sum_{ij} |\cdot|$, which is the lasso penalty that encourages few edges to change between subsequent time points while the rest of the structure remains the same (Danaher, Wang and Witten, 2014).
- $\Psi(\cdot) = \ell_{12}(\cdot) = \sum_j \| \cdot_j \|_2$, iwhich s the group lasso penalty that encourages the graph to restructure at some time points and to stay stable in others (Gibberd and Roy, 2017; Hallac, Leskovec and Boyd, 2015).
- $\Psi(\cdot) = \ell_2^2(\cdot) = \sum_{ij} (\cdot_{ij})^2$, which is the Laplacian penalty which encourages smooth transitions over time, for slow changes of the global structure (Weinberger et al., 2007).
- $\Psi(\cdot) = \ell_\infty(\cdot) = \sum_j (\max_i |\cdot_{ij}|)$, which is the max norm penalty which encourages a block of nodes to change their structure with no additional penalty with respect to the change of a single edge among such nodes.
- $\Psi(\cdot) = \min_{V: A = V + V^\top} \sum_j \|V_j\|_p$, which is the row-column overlap penalty that encourages a major change of the network at a specific time, while the rest of the system is enforced to remain constant. Choosing $p = 2$ causes the penalty to be node-based, *i.e.*, the penalty allows for a perturbation of a restricted number of nodes (Mohan et al., 2012).

1.8 Summary

In this chapter we provided background on graphical models inference methods based on various probability distribution assumptions. In particular, we showed how it can be possible to exploit a penalised Maximum Likelihood

Estimation strategy to infer a graphical model that assumes a distribution belonging to the Exponential Family class. We described in details the inference for Gaussian Graphical Models, Ising Graphical Models and Poisson Graphical Models. Lastly, we discuss the state-of-the-art temporal extensions to these models focusing on the one that extends GGMs with temporal consistency.
We want to remark that it would be interesting to consider in the future other distributions as the Multinomial (Yang et al., 2013) or the exponential (Yang et al., 2015) as well as combinations of all the distributions (Lee and Hastie, 2015; Yang et al., 2014; Žitnik and Zupan, 2015).

2

Gaussian Graphical Models with Missing Data

Whenever analysing real-world systems it would be appropriate to consider missing data in order to devise a non-biased analysis and therefore end up with better estimate of the statistical model parameters. Indeed, it is likely to get incomplete observations of all the variables acting in a phenomenon. Of the possible types of missing observations we can distinguish two cases: *latent* and *partial* variables (Little and Rubin, 2019).
Latent variables are defined as un-observed variables that need to be inferred from observed ones. Since we cannot measure them we may not know how they influence the system. As an example, such variables could be experimental conditions that were not taken into account during the measurement of the system. Partial variables, instead, are those variables for which some observation is missing but we are still able to partially detect their behaviour. Therefore, we can analyse them but we may incur in problems related to the holes in the dataset. Note that latent variables are a degeneration of partial variables where none of the observations is present.
As previously introduced, the inference of graphical models from observations is a difficult task as the solution lies in a combinatorial space. For this reason inference methods assume that the underlying graph is sparse to improve its identifiability (see Section 1.3.1). The presence of latent and/or partial variables makes such inference even a harder task as partial variables convey bias in the estimated edges, while, the presence of latent variables leads to locally dense structure in the graph. Given the assumption of sparse graphs this induces an identifiability problem since there are infinitely many possible marginalisation on the graph that induce the same dense structure.

In this thesis, we focus on methods that estimate Gaussian Graphical Models (GGMs) when data may be partial or latent (Chandrasekaran, Parrilo and Willsky, 2010; Städler and Bühlmann, 2012; Yuan, 2012). We restrict ourselves to the Gaussian case as it easily allows to include missing data assumptions in the model.

OUTLINE This chapter is organised as follows. In Section 2.1 we introduce the concept of missing data. In Section 2.2 we describe in general terms the Expectation-Maximization algorithm widely used in partial and latent variables contexts. In Section 2.3 we present a method based on EM for the inference of Graphical Lasso with partial data while in Section 2.4 we present two algorithms for the inference of a network in presence of latent data. We conclude with Section 2.5, a summary of the chapter.

2.1 Missing data

Definition 12 (Missing data)**.** Missing data are un-observed values that would be meaningful for the analysis if observed: a missing value might hide a meaningful value.

Missing data can be of different types, we depicted the ones we are interested to deal with in Figure 3. We identify missing values with dashed squares. In particular we focus on panel (b) that shows a dataset in which we have factored data (Little and Rubin, 2019), *i.e.*, the variable X_3 is never observed and, we will call it from now on latent. And also panel (c) in which we have randomly positioned missing values.
In the analysis of these types of data being aware of the mechanism that induced missing data is crucial in order to better develop analysis method and interpret the results. In this thesis we will consider data in which the mechanism that induces missing values is *ignorable* (Little and Rubin, 2019). More specifically, we consider data that are *Missing At Random*.
Suppose we have N observations of D variables $(X_1, \dots, X_D) \sim \mathcal{N}(0, \Sigma)$ in a data set $X \in \mathbb{R}^{N \times D}$ and, for each sample i, two sets of indices $\mathbb{I}_{M_i} = \{1, \dots, M_i\}$ and $\mathbb{I}_{O_i} = \{M_i + 1, \dots, D\}$. We denote with $X[i,:] = (X[i, O_i],\ X[i, M_i])$ the i-th random sample for $i = 1, \dots, N$ and with

$$X[O] = (X[1, O_1], X[2, O_2], \dots, X[N, O_N])$$

the set of observed variables across all samples. The notation is similar for $X[M]$.

Definition 13 (Missing At Random)**.** The variables are Missing At Random (MAR) if observed values are a random sub-sample of the sampled values, in this case the mechanism is ignorable. On the contrary, if the probability that $X[iv]$ is observed depends on the value of $X[iv]$ then the missing-data mechanism is non-ignorable.

Let now $p_G(X|K) = p_G(X[M], X[O]|K)$ be the density function of the joint distribution over the parameters K. In a graphical model problem, p factorises according to a graph G and the parameters K denote the adjacency matrix of the graph. The marginal probability on the observed variables can be computed summing the missing values

$$p_G(X[O]|K) = \int p_G(X[M], X[O]|K) dX[M].$$

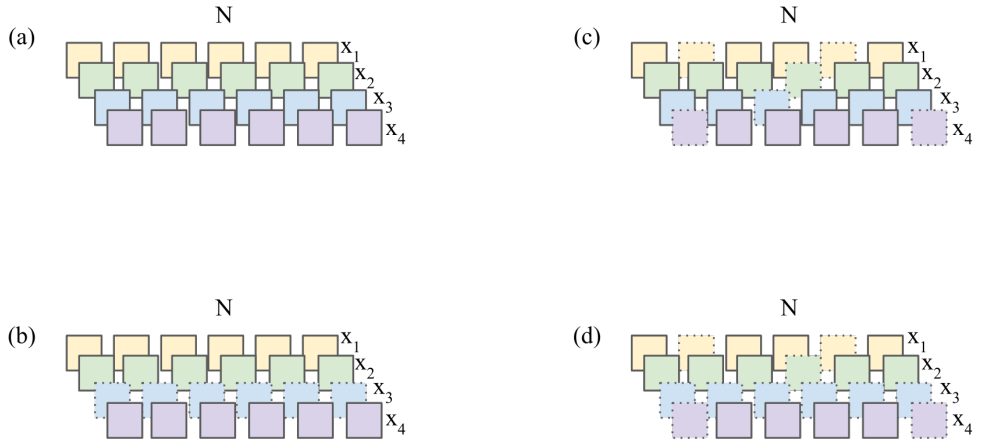


FIGURE 3. *Graphical representation of a dataset with different possible conditions of missing values: (a) all the observations for all the variables are available; (b) all the observations for X_3 are missing* (i.e., *X_3 is latent); (c) some values missing completely at random Little and Rubin, 2019; (d) all observations for one variables are missing and the other variables have some missing observations.*

Theorem 1 (6.1A Little and Rubin, 2019)**.** Let K be the parameters of the graphical model p_G and ψ the unknown generative mechanism of missing data. The latter is ignorable for the inference of K if the following two conditions hold:

- the parameters K and ψ are distinct, in the sense that the joint parameter space (K, ψ) is the product of the parameter spaces.
- the full likelihood factorises as

$$L(X, R|K, \psi) = L(X[O]IK)L(X, R|\psi)$$

where R is an indicator function defined as

$$R(i, v) = \begin{cases} 1 & \text{if } X[iv] \text{ is observed} \\ 0 & \text{if } X[iv] \text{ is missing} \end{cases}$$

Given the theorem the MLE of the parameters K can be obtained by maximising the marginal likelihood on the observed data, defined as

$$L(X[O]|K) \propto p_G(X[O]|K).$$

Hence, the MLE of the parameters K on complete or only observed data is equivalent, as it is based on the marginal likelihood provided that the missing data mechanism can be ignored.

DIFFERENCE BETWEEN PARTIAL AND LATENT VARIABLES The formal difference between partial and latent variables lies in how the set of indices $\mathbb{I}_M$ and $\mathbb{I}_O$ are defined. Indeed, if data are partial these sets change for each sample as we would expect that we have random missing values that change their position across samples. If the data are latent, instead, the sets $\mathbb{I}_M$ and $\mathbb{I}_O$ are stable across samples, *i.e.*, the variables in the set $\mathbb{I}_M$ are never observed and, therefore, they may not be comprised in the input data matrix.

2.2 Expectation Maximization Algorithm

The modelling of data through statistical methods involve the inference of unknown parameters from observations. As an example we may want to infer the mean and the variance of a univariate normal distribution. In some contexts the unknown parameters may be coupled with missing values in the data. The *Expectation Maximisation* (EM) algorithm is an iterative method to find Maximum Likelihood or a posterior estimates in these statistical models where there are two types of unknowns (Dempster, Laird and Rubin, 1977). The two sets of unkowns are the parameters K and the missing variables $X[M]$. Given these unknowns the MLE becomes a set of intertwined equations in which the inference of the parameters requires the values of the missing variables and vice-versa.
The EM algorithm is based on a simple idea: we can pick arbitrary values for one of the two sets of unknowns (either for K or for $X[M]$) and use them to estimate the second set, then use these new values to find a better estimate of the first set, and keep alternating between the two until both the resulting values converge to fixed points (Dempster, Laird and Rubin, 1977). This translates into the alternation of two steps:

- the **Expectation (E) step**, in which we estimate the missing data based on the likelihood of the available data and the parameters K at the previous iteration;
- the **Maximization (M) step**, in which we maximize the likelihood given the complete data estimated during the E step.

More formally, given the statistical model, a set $X[O]$ of observed data, a set of missing values $X[M]$, a vector/matrix of unknown parameters K and a likelihood function

$$L(X[O], X[M]|K) \propto p_G(X[M], X[O]|K)$$

the MLE of the unknown parameters is determined by maximizing the marginal likelihood of the observed data

$$L(X[O]|K) = p_G(X[O]|K) = \int p(X[O], X[M]|K) dX[M].$$

Nevertheless, this quantity is often intractable. Therefore, EM algorithm seeks to find the MLE of the marginal likelihood with two steps:

EXPECTATION STEP (E STEP) We denote with $Q(K|X[O], K^\iota)$ the expected value of the log likelihood function of K with respect to the current conditional distribution of $X[M]$ given $X[O]$ and the current estimates of the parameters K^ι.

$$Q(K|X[O], K^\iota) = \mathbb{E}_{X[M]|X[O],K^\iota} \left[\ell(K|X[O], X[M]) \right].$$

Algorithm 1 EM algorithm

1: $\theta \leftarrow$ initialise values
2: **repeat**
3: $\quad p_M^k \leftarrow (X_M|\theta)$ // compute the probabilities of the missing variables
4: $\quad \theta^k \leftarrow$ maximize $L(X_M|\theta)$ // maximise parameters given the probabilities
5:
6: **until** convergence

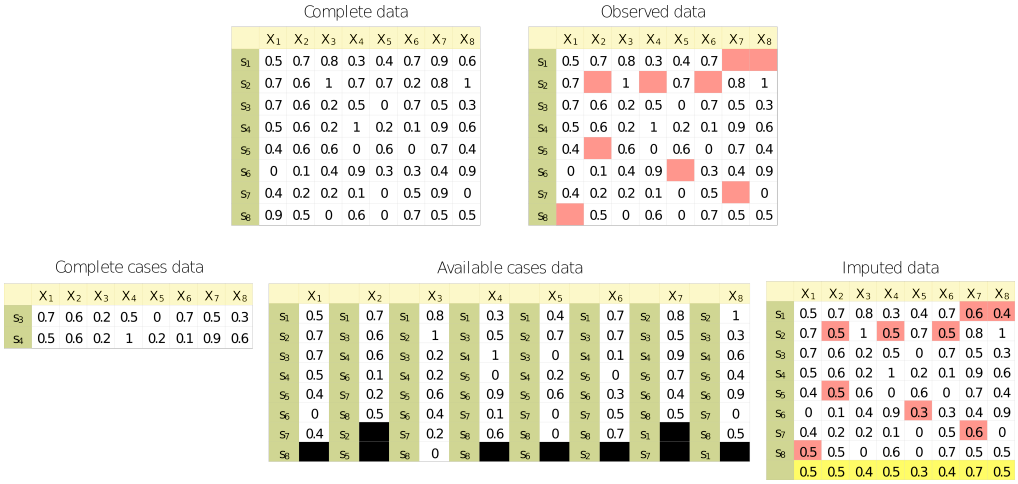


FIGURE 4. *Toy example of different initialisation strategies to deal with partial data. In the top left corner we have the original complete data matrix and on its right the one that we can actually observe. On the bottom we can see the results obtained with complete cases (left) in which we reduced the samples size to two available samples, available cases (middle) in which we have different samples for each variable and imputing (right) in which we insert an empirical mean different from the true one (bottom yellow row).*

MAXIMIZATION STEP (M STEP) Given $Q(K|X[O], K^\iota)$ we find the parameters K that maximize such quantity as

$$K^{(\iota+1)} = \underset{K}{\operatorname{argmax}}\, Q(K|X[O], K^\iota).$$

This translates into a simple algorithm presented in Algorithm 1 which can be proved to reach a local minimum of the cost function (Dempster, Laird and Rubin, 1977).
In some cases EM can be used to compute iterative MLE by *creating* fully missing variables in such a way that in M step is non-iterative.

2.2.1 *Initalisations*

The very first step of EM (Algorithm 1) is the initialisation of the values which can be performed with different strategies when we have missing data (Little and Rubin, 2019). We reported a visual representation in Figure 4 that illustrates the various initialisation methods. With partial variables we can use:

1. **Complete cases**. We restrict the analysis to the cases where all the D variables are present. This approach is simple and allows comparability of statistics but leads to potential loss of information or bias as the complete cases are a random sub-sample of the original cases.

2. **Available cases**. It uses all the available information by including all the cases where the variable of interest is present. Its main disadvantage is the explicit difference in each variable that depends on the pattern of missing data. In GGMs inference this could lead to non-positive definite covariance matrices. An evolution is the *pairwise available-cases* method that uses samples based on the co-presence of pairs of variables.

3. **Imputing**. It consists in inserting some heuristics in place of the missing values in the input data matrix. There are different techniques to compute the heuristic value to put as the mean, the mode or the most frequent of the available values.

In presence of latent variables we can randomly generate data in place of the missing ones under the constraints specified by the problem. This could lead to reach bad local minima in the optimisation function but always ensure the problem to be well defined. Such approach is suitable also in the case of partial data.

2.3 GGMs with Partial Data

In high dimensional contexts the problem of inferring a GGMs has been widely studied (Friedman, Hastie and Tibshirani, 2008; Ravikumar et al., 2009a; Wainwright, Lafferty and Ravikumar, 2007). Nontheless, it is crucial to consider that often datasets contain missing values (Little and Rubin, 2019). The estimate of mean values and covariance matrices becomes difficult when the data is incomplete and no explicit maximisation of the likelihood is possible. In (Städler and Bühlmann, 2012) the authors proposed a method for estimating the inverse covariance matrix in a high-dimensional multivariate normal models in presence of partial data. This method allows for the inference of a graph structure while supporting imputation on the original data matrix.
A simple way to estimate the adjacency matrix K, that in GGMs corresponds to the precision matrix, is to delete all the cases containing missing values and then estimating the covariance by solving the graphical lasso (GL) problem using only the complete cases (Friedman, Hastie and Tibshirani, 2008). However, the exclusion of cases can result in a substantial decrease of the sample size available and to a consequent bias, especially when $D \gg N$. Another possible method is imputing the missing values with the corresponding mean and then solving the GL.
While both these approaches are effective it is shown that they work more poorly than the following method called *Missing Graphical Lasso* (MissGL) (Städler and Bühlmann, 2012). This method is based on the inference of the precision matrix K by maximising the observed log-likelihood ℓ_{GGM} (Eq. (11)), under the

assumption that the underlying missing data mechanism is ignorable. Assuming $D \gg N$ the functional to maximise to estimate $K \in \mathcal{S}_{++}^{D}$ is

$$\underset{K \succ 0}{\text{minimise}} - \ell_{GGM}(X[O]|K) + \lambda \|K\|_1 \tag{20}$$

which, despite the concise appearance, tends to be a complicated non-convex functional for any general missing data pattern with the possible existence of multiple stationary points (Schafer, 1997; Städler and Bühlmann, 2012).
The optimisation of such model can be performed through EM algorithm following the steps in Algorithm 1. In particular the maximisation step translates into the optimisation of the Graphical Lasso (GL) problem with the estimated complete data empirical covariance matrix.
To derive the EM algorithm we note that the complete data X follows a multivariate normal distribution that belongs to the regular exponential family with sufficient statistics

$$\mu^C = X^\top I_D = (\sum_{i=1}^{N} X_{i1}, \sum_{i=1}^{N} X_{i2}, \ldots, \sum_{i=1}^{N} X_{iD}) \tag{21}$$

$$C = (X^\top X) = \begin{pmatrix} \sum_{i=1}^{N} X_{i1}^2 & \sum_{i=1}^{N} X_{i1}X_{i2} & \ldots & \sum_{i=1}^{N} X_{i1}X_{iD} \\ \sum_{i=1}^{N} X_{i2}X_{i1} & \sum_{i=1}^{N} X_{i2}^2 & \ldots & \sum_{i=1}^{N} X_{i2}X_{iD} \\ \vdots & \vdots & & \vdots \\ \sum_{i=1}^{N} X_{iD}X_{i1} & \sum_{i=1}^{N} X_{iD}X_{i2} & \ldots & \sum_{i=1}^{N} X_{iD}^2 \end{pmatrix} \tag{22}$$

where μ^C is the empirical sample mean, C is the empirical covariance matrix and the matrix I_D denotes the identity matrix of dimension $D \times D$ (Städler and Bühlmann, 2012). Note that, differently from other network inference case, we cannot assume the mean to be 0. Indeed, while with the graphical lasso problem on complete data we would re-centre the data using the empirical mean, given the holes in our dataset doing so would convey bias. Therefore, we are forced to infer the means as well.
The complete data log-likelihood can be expressed in term of the sufficient statistics as

$$\ell_{GGM}^{c} = -\frac{N}{2}\log\det(K) + \frac{1}{2}\operatorname{tr}(KC) + \frac{N}{2}\mu^\top K\mu - \mu^\top K\mu^C + \lambda\|K\|_1$$

Therefore the E step at the ι-th iteration of the EM algorithm estimates the sufficient statistics by filling the holes of the dataset with their expected value. In particular we compute:

$$\mathbb{E}[X[iv]|X[iO_i], K^\iota, \mu^\iota] = \begin{cases} X[iv] & \text{if } X[iv] \text{ both observed} \\ c_i[v] & \text{if missing} \end{cases}$$

where the vector $c_i \in \mathbb{R}^{|M_i|}$ for $i = 1, \ldots, N$ are the values that we substitute in place of the missing values. In particular, a good value to substitute is the

expected mean of each variables. In order to find it we observe that the missing values are still distributed normally in the following way

$$X[i, M_i]|X[i, O_i] = \mathcal{N}\Big(\mu[M_i] + K[M_i]^{-1}K[M_iO_i](X[O_i] - \mu[O_i]), K[M_i]^{-1}\Big).$$

Therefore,

$$c_i = \mu^\iota[M_i] - (K[M_i]^\iota)^{-1}K[M_i, O_i]^\iota(X[i, M_i] - \mu[O_i]^\iota)$$

Similarly, we compute the expectation for the second sufficient statistics as

$$\mathbb{E}[X[iv]X[iv']|X[iO_i], K^\iota] = \begin{cases} X[iv]X[iv'] & \text{if both observed} \\ X[iv]c_i[v'] & \text{if } X[iv'] \text{ missing} \\ (K[v, v']^\iota)^{-1} + c_i[v]c_i[v'] & \text{if both missing} \end{cases}$$

Missing values are thus replaced by the conditional mean given the set of values observed for that observation. These conditional means and the non-zero conditional covariances are easily found from the current parameters estimates. The M step is straightforward and consists in optimising the GL to obtain the precision matrix K and to simply take the mean of the computed sufficient statistic μ^C to compute the mean μ. We obtain the means as

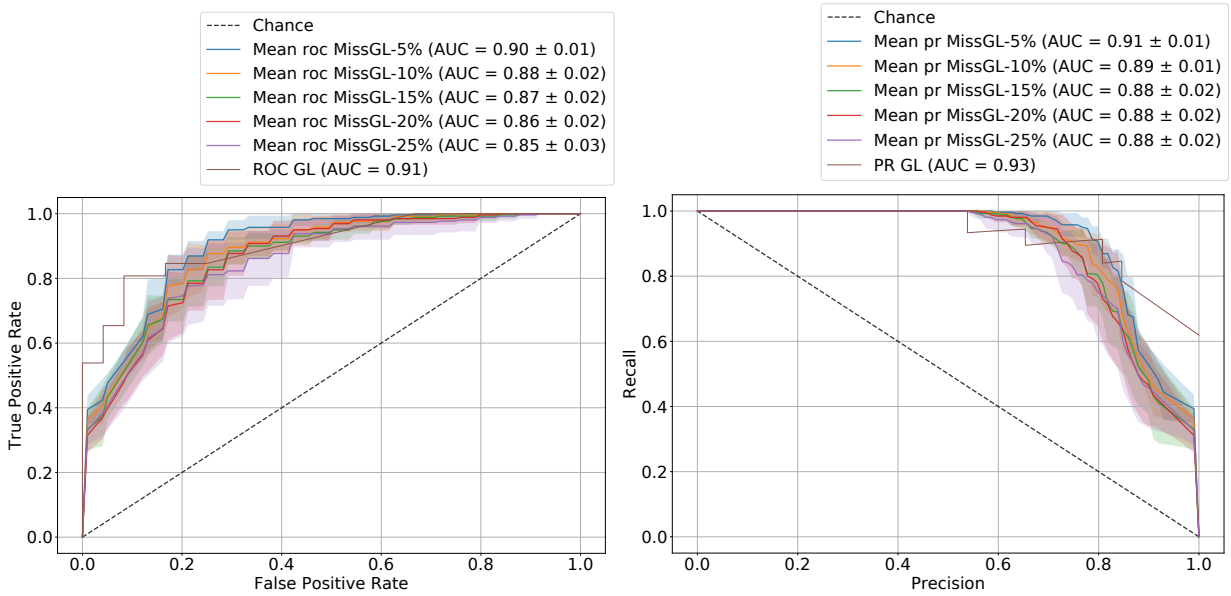
$$\mu^{\iota+1} = \frac{1}{N}(\mu^C)^{\iota+1}$$

while for the estimate of $K^{\iota+1}$ we solve the GL problem (Equation (12)) with empirical covariance matrix on the complete data defined as

$$C[vv']^{\iota+1} = \sum_{i=1}^{N} \mathbb{E}[X[iv]X[iv']|X[iO], K^\iota]. \tag{23}$$

2.3.1 *Synthetic Data Experiments*

We wanted to show the efficacy of the proposed method compared to the graphical lasso (GL) (Friedman, Hastie and Tibshirani, 2008). In particular, we generated a network of $D = 10$ nodes each of them connected with at most 3 other nodes in the network following the Barabasi-Albert random graph generation provided by the NetworkX python package (Albert and Barabási, 2002; Hagberg, Swart and S Chult, 2008). We randomly generated with uniform distribution weights for the edges in the interval $[-1, 1]$ and we then sampled $N = 100$ from the corresponding multivariate normal distribution.
We applied GL and Miss-GL on the data. To simulate missing data we removed the $5, 10, 15, 20, 25\%$ of the data for 10 times. We used a fixed hyper-parameter α on all the inference methods to better compare the results that are shown in Figure 5. We observe that while GL always performs better than MissGL the difference is not so significative. Also, MissGL is stable under an increasing percentage of missing values which make it suitable for large set analysis.



(A) *ROC curves* (B) *Precision-Recall curves*

FIGURE 5. *Average results across 10 repetitions for the comparison in terms of ROC and PR curves of the Graphical Lasso (GL) and the Missing Graphical Lasso (MissGL)on a dataset of $D = 10$ nodes and $N = 100$ samples. MissGL is applied on the dataset at different percentages of random missing values.*

2.4 GGMs with Latent Data

Latent variables cause observations to do not correspond to what we expect. Indeed, by externally influencing the system they cause a non-sparse graph were spurious dependencies between observed variables are introduced (Choi, Chandrasekaran and Willsky, 2009; Choi et al., 2011).
In Figure 6 we report a toy example of the effect of the latent variables on the inferred network. We can observe that the true network on the observed variables (middle one) has less edges than the one that does not consider latent variables (right one). This is due to the fact that connections existing with the latent variables are forced to be explained by the observed variables only, thus introducing spurious links (dashed one in right network).
Methods for the inference of GGMs can be extended in order to consider latent variables able to represent factors which are not observed in the data. Note that, these latent variables are not principal components, since they do not provide a low-rank approximation of the graphical model. On the contrary, such factors are added to the model in order to condition the statistics of the observed variables. In particular, one can consider both latent and observed variables to have a common domain (Choi et al., 2011).
Given N random samples $X_1, \dots, X_N$ of length $|O| \leq D$, these observations can be viewed as the first components of a random sample $X[i,:] = (X[iO], X[iM])$ of a multivariate distribution $\mathcal{N}(\mu, \Sigma)$ such that $X[iO]$ are the observed data and $(X[iO], X[iM])$ are the complete data both observed and latent. Note that

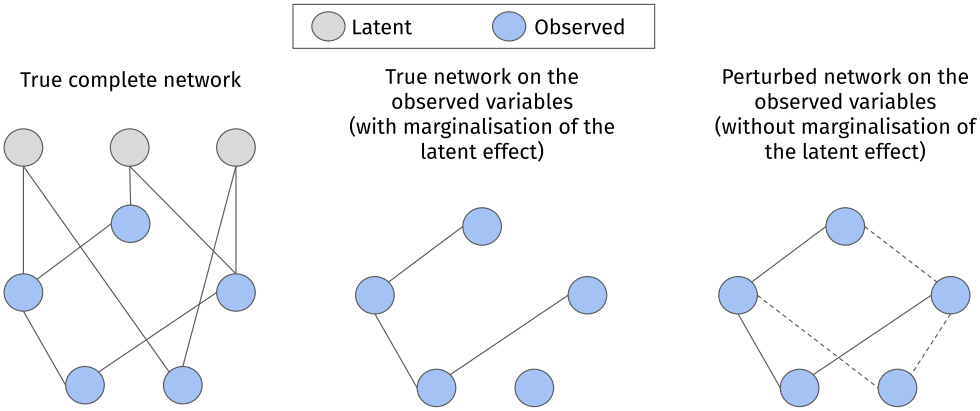


FIGURE 6. *Toy example of a network structure when we consider the latent variables. In the leftmost network we see the complete network on all the variables, both latent and observed, in the middle the true network only on the observed variables while in the rightmost network we have the network on the observed variables if we do not consider the influence of the latent.*

the set of missing values denoted by the set $\mathbb{I}_M$ does not vary with samples but remains stable such that $M = M_1 = M_2 = \cdots = M_N$ and $O = O_1 = O_2 = \cdots = O_N$, *i.e.* the missing values are patterned into a group of variables that are never observed. In this case we can assume the mean μ to be zero without introducing bias in the analysis and, thus, semplifying the inference. The distribution of such block data depends on a covariance (or its inverse the precision) matrix, that can be represented as

$$\Sigma = \left[\begin{array}{c|c} \Sigma[M] & \Sigma[MO] \\ \hline \Sigma[OM] & \Sigma[O] \end{array}\right]$$

Similarly, we can define the precision matrix K of the joint distribution of both latent and observed variables partitioned into four blocks:

$$K = \Sigma^{-1} = \left[\begin{array}{c|c} K[M] & K[MO] \\ \hline K[OM] & K[O] \end{array}\right]. \tag{24}$$

Both matrices lie in $\mathcal{S}^D_{++}$ and their blocks represent the conditional dependencies among latent variables ($K[M]$), observed variables ($K[O]$), between latent and observed ($K[MO]$), and vice-versa ($K[OM]$).

The distribution of the missing and observed part are then conditioned on the other as follows

$$X[M]|X[O] = \mathcal{N}\Big(K[M]^{-1}K[MO]X[O], K[M]^{-1}\Big)$$

and similarly

$$X[O]|X[M] = \mathcal{N}\Big(K[O]^{-1}K[OM]X[M], K[O]^{-1}\Big)$$

The marginal precision matrix of the observed variables is then given by the Schur complement w.r.t. the block K_M (Chandrasekaran, Parrilo and Willsky, 2010; Horn and Johnson, 2012):

$$\hat{K}[O] = K[O] - K[OM]K[M]^{-1}K[MO] = K[O] - L. \tag{25}$$

The estimation of a graphical model with these type of data is more difficult as there are identifiability issues. Indeed, there are possibly infinite marginalisation of latent variables that lead to the same sparse estimate of $K[O]$.
In order to infer the GGM is possible to proceed in two ways: by estimating the latent variables through an EM approach that provides a complete estimate of K but, in turn, is a non-convex procedure or, by exploiting the Schur complement form and estimating a marginalisation of the latent variables that does not provide a complete estimate of K. This latter approach has the major advantage of being a convex problem.

2.4.1 *Non-Convex Approach*

The method proposed for the inference of GGMs with partial data can be used also for latent data as they are a partial data particular case. The same idea proposed in (Städler and Bühlmann, 2012) for partial data has been proposed separately for latent data in (Yuan, 2012) but it is, indeed, the same thing. We call such method *Latent Variable Graphical Lasso* (LVGLASSO) (Yuan, 2012). LVGLASSO exploits the EM algorithm to minimise directly the problem presented in Equation (20). The EM provides an estimate of the complete data covariance matrix allowing for the inference of a network also on the latent part of the precision matrix K.
The procedure, similarly to the one for partial data infers the global precision matrix by estimating the latent part. Then, similarly to MissGL, after the expectation step the problem translates in a standard graphical lasso problem (Friedman, Hastie and Tibshirani, 2008). Let

$$C = \frac{1}{N}(X[O], X[M])^\top (X[O], X[M])$$

be the complete data sufficient statistic and

$$C[O] = \frac{1}{N}(X[O])^\top (X[O])$$

Algorithm 2 EM-LGL

1: **for** $\iota = 1, \dots$ **do**
2: $\quad C[OM] = X[O]^\top X[O] K[OM] K[M]^{-1}$
3: $\quad C[M] = K[M]^{-1} + (K[M]^{-1} K[MO] (X[O]^\top X[O]) K[OM] K[M]^{-1})$
4: $\quad C = \left[\begin{array}{c|c} C[M] & C[OM]^\top \\ \hline C[OM] & C[O] \end{array}\right]$.
5: $\quad K^\iota = \underset{K \succ 0}{\operatorname{argmin}} \operatorname{tr}(CK) - \log\det(K) + \alpha \|K\|_1$
return K

the observed sufficient statistic (empirical covariance matrix). Complete data statistic can be estimated in the same way described in Equation (23). As we assume the mean to be zero, we ignore the μ^C sufficient statistic.
The related problem is non-convex. Therefore, Algorithm 2 may reach local optima dependent on the initialisation of the initial parameters. Nevertheless, LVGLASSO presents good performance in terms of structure recovery whilst opening the road for the estimate of the latent variables themselves.

2.4.2 *Convex Approach*

A regularised convex approach for GGM selection with latent variables, namely *Latent Graphical Lasso* (LGL), was proposed in (Chandrasekaran, Parrilo and Willsky, 2010). The intuition of this method lies in the decomposition of the precision matrix of the marginal distribution on the observe variables sparse component plus a low-rank component. These two parts are regularised separately with an ℓ_1 norm and a nuclear penalty.
The two main assumptions of LGL for the identifiability of the two matrices are complex and derive from theoretical evaluations in (Chandrasekaran et al., 2011).

1. The rank r of the matrix L is the number of latent variables, by definition. In order to being able to identify such matrix we need to assume that $r = |M| \ll |O|$, *i.e.*, there are few latent variables compared to the number of observed ones.

2. The effect of the marginalisation is scattered over many observed variables. In other words, in the network the latent variables are conditionally dependent with the majority of the observed ones. This is fundamental to not confound their marginalisation with the true underlying conditional sparse structure of $K[O]$.

The two parts, corresponding to the two parts of the Schur complement in Equation (25), specify respectively the *conditional statistics* ($K[O]$) and a summary of the marginalisation effect over the latent variables (L). Typically the

conditioned $\hat{K}[O]$ is not sparse, but the subtraction (marginalisation) of the latent factors contribution allows for the recovering of the true sparse GGM.
The LGL model is defined by the following functional (Chandrasekaran, Parrilo and Willsky, 2010; Ma, Xue and Zou, 2013)

$$\begin{aligned} \underset{\substack{(K,L) \\ K\succ 0, L\succeq 0 \\ \text{rank}(L)\leq r}}{\text{minimise}} \quad & -\ell_{GGM}(X[O]|K-L) + \lambda(\gamma\|K\|_1). \end{aligned}$$

Such functional, though, is non-convex because of the rank constraint. It is possible to relax such constraint by imposing a nuclear norm as follows

$$\begin{aligned} \underset{\substack{(K,L) \\ K\succ 0, L\succeq 0}}{\text{minimise}} \quad & -\ell_{GGM}(X[O]|K-L) + \lambda(\gamma\|K\|_1 + \|L\|_*), \end{aligned}$$

where

$$\|\cdot\|_* = \ell_*(\cdot) = \text{tr}(\sqrt{\cdot^T\cdot}) = \sum_{i=1}^{D}\sigma_i(\cdot) \tag{26}$$

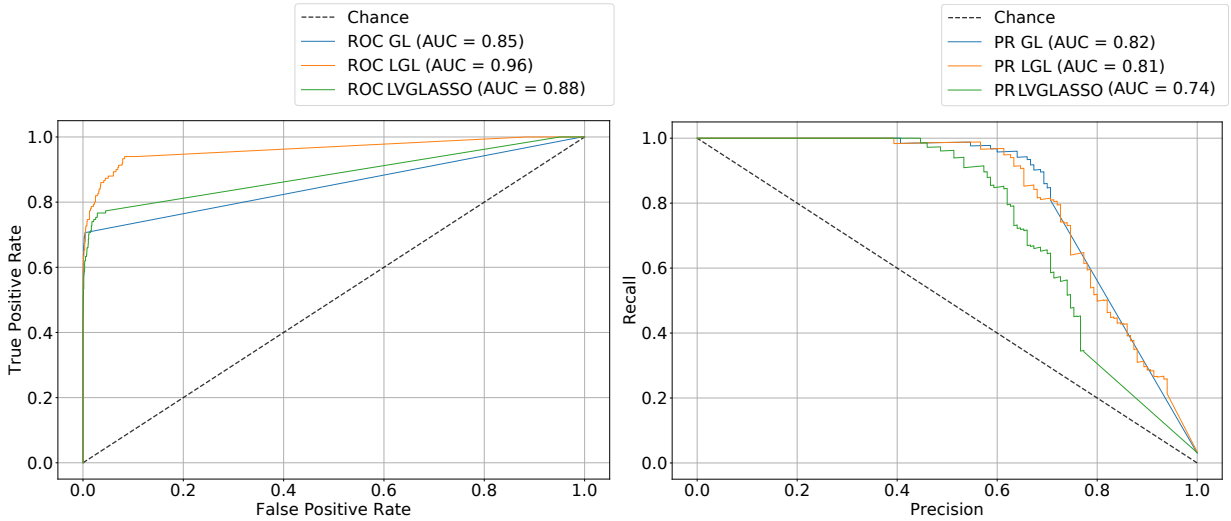
where each σ_i denote a singular value. While not being a strong constraint this norm encourages the matrix L to have small rank. For this model, with a suitable choice of λ, there exist a range of values of γ for which the estimates (K, L) have, with high probability, the same sparsity and sign pattern and rank as $K^*[OM](K^*[M])^{-1}K^*[MO]$ (Chandrasekaran, Parrilo and Willsky, 2010, Theorem 4.1).
The functional, for simplification of the optimisation algorithm, can be also rewritten as (Ma, Xue and Zou, 2013):

$$\begin{aligned} \underset{\substack{(K,L) \\ K\succ 0, L\succeq 0}}{\text{minimise}} \quad & -\ell_{GGM}(X[O]|K-L) + \alpha\|K\|_{od,1} + \tau\|L\|_*, \end{aligned}$$

In Chapter 5 we rely on this last form which allows to model the two penalties (ℓ_1- and ℓ_*) separately.

2.4.3 *Synthetic Data Experiments*

We want to show the effectiveness of LGL and LVGLASSO compared to the baseline Graphical Lasso (GL) (Friedman, Hastie and Tibshirani, 2008). We generated a network on $|O| = 100$ observed variables and $|M| = 5$ latent from which we sampled $N = 100$ observations.
We applied GL, LGL and LVGLASSO on data changing the hyper-parameters as the model changes. Results, in the form of ROC and PR curves, are shown in Figure 7. In particular in Figure 7a and Figure 7b we observe that LGL is the method that performs best, while LVGLASSO and GL have similar performances. This is due to the fact that while both LGL and LVGLASSO consider latent variables the second is non-convex and therefore it may require multiple initialisations to obtain the best result. Nonetheless it offers an estimate of the latent variables impossible to obtain with LGL.



(A) *ROC curves* (B) *Precision-Recall curves*

FIGURE 7. *Results comparison in terms of ROC and PR curves of the Graphical Lasso (GL), the Latent Graphical Lasso (LGL) and the Latent Variables Graphical Lasso (LVGLASSO) on the inferred observed part of the adjacency matrix of a graph with* $|O| = 100$ *observed variables,* $|M| = 5$ *latent and* $N = 100$ *samples.*

2.5 Summary

In this chapter we provided a background knowledge on the concept of missing data and the possible declinations into partial and latent variables. We showed the inference methods that can be used to infer GGMs that cope with these type of data. We exploited regularised optimisation methods . In literature we can find other methods that deal with latent data as (Beal and Ghahramani, 2003) that finds a structural EM approach to estimate latent variables in Bayesian Networks (Friedman, 1998) as well as latent variables in a Baysian dynamic model (Beal et al., 2004). Li, Jia and Yao, 2015 estimate Gaussian latent variables from observed categorical variables, while Robin, Ambroise and Robin, 2018 perform EM assuming a tree structure between latent and observed nodes. Also, Fan et al., 2019 adopt an ADMM optimization procedure to infer a precision matrix with noisy and missing data.
Here, we focused on the Gaussian distribution as it has nice property that simplify the inference process also in presence of missing data. Methods that deal with missing data also assuming other distributions are present in literature (Anandkumar et al., 2013; Anandkumar et al., 2012; Schwing et al., 2012). We are aware of their importance but we leave their temporal extension to future work.

PART II

Contribution

Part 2 describes the original contribution of this thesis. In particular in Chapter 3 we introduce the main measures and model selection techniques in the context of network inference methods together with our extension for stability-based model selection. Chapter 4 describes the generalised models for Markov models with the possibility to impose different probability distribution assumptions as well as different pattern of time dependency. Chapter 5 presents the temporal extension to models with missing data variables and, lastly, Chapter 6 describes the Python framework containing the code of this thesis as well as other utilities and algorithms.

3

Hyper-parameters Selection and Performance Evaluation

Part of this chapter content is present in the following publications:
Veronica Tozzo, and Annalisa Barla. Multi-parameters Model Selection for Network Inference. International Conference on Complex Networks and Their Applications. Springer, Cham. (2019)

Methods for the inference of graphical models allow to consider a large variety of different real-world data. These methods lie in the unsupervised machine learning category, for which two questions easily arise: how do we measure the goodness of a new proposed model? How do we select the hyper-parameters that are suitable for a specific dataset? In this chapter we try to address this two problems. As for the first question, in literature possible outlines for a rigorous analysis ar provided. They often rely on evaluating the model on synthetic data that are simulated as close as possible to the real-data we aim at modelling. The answer to the second question is more challenging given the unsupervised nature of the problem. Indeed, while many methods for model selection exist in the supervised settings this is not always true in the unsupervised setting case. In particular one may rely on model selection based on likelihood scores or on the stability of the solution. While the first one is easily extendible to our main case study (the multi-parameter multi-networks case), the second requires some further consideration.

OUTLINE This chapter is organised as follows: in Section 3.1 we formalise a generic form for a multi-network multi-penalty inference functional; Section 3.2 we present the metrics used throughout the thesis to measure the performances of inference methods on synthetic data. In Section 3.3 we present our generalisation of model selection criteria for network inference methods. We conclude in Section 3.4 with a summary of the chapter and future research directions.

3.1 General network inference functional

Probability-based multiple network inference aims at estimating T graphs $G_t = (V, E_t)$ for $t = 1, \dots, T$ where $V = \{1, \dots, D\}$ are the nodes and $E_t \subseteq V \times V$ is the set of edges that connect such nodes in the network t. The inference of the weighted adjacency matrices of such graphs $\boldsymbol{K} = (K_1, \dots, K_T)$ is performed from observations $\boldsymbol{X} = (X_1, \dots, X_T) \in \mathbb{R}^{N_1 \times D} \times \cdots \times \mathbb{R}^{N_T \times D}$. We define a generic form for the inference problem as

$$\underset{\boldsymbol{K}, K_t \succ 0}{\text{minimize}} \sum_{t=1}^{T} \left[-\ell(X_t|K_t) + \alpha \|K_t\|_{1,od} \right] + \sum_{p=1}^{P} \beta_p \mathcal{P}_p(K_1, \dots, K_T) \tag{27}$$

where $\|K_t\|_{1,od}$ is the off-diagonal ℓ_1 norm that enforces sparsity on the off-diagonal elements of each adjacency matrix K_t and $\mathcal{P}_p$ is typically a sum of penalties, controlled by the hyper-parameter β_p, applied on combinations of the precision matrices. The main hyper-parameter, α, regulates the sparsity of the solution, a fundamental assumption to reduce the complexity of the problem at hand.
Such functional is associated with a solver denoted with ζ, *i.e.*, an optimisation algorithm . Such algorithm ζ takes in input the set of matrices $\boldsymbol{X}$ and the hyper-parameters of the model. It returns as output the set of adjacency matrix of the graph $\boldsymbol{K}$, that will depend, therefore, on the input hyper-parameters.
We can provide examples of instantiation of the general functional in the case of GGMs. Indeed, in this chapter, in order to assess the reliability of stability-based model selection criteria against likelihood-based we will use GGMs as they allow to compute the joint likelihood of the model. In this case, data are assumed to be sampled from a multivariate normal distribution and each graph G_t is inferred from samples $X_t \in \mathbb{R}^{N_t \times D} \sim \mathcal{N}(0, K_t^{-1})$ If we substitute ℓ_{GGM} in the functional (27) we can obtain different multiple GGMs:

- by taking $T = 1$ and $P = 0$ Equation (27) has the same form of the standard Graphical Lasso problem (Friedman, Hastie and Tibshirani, 2008);
- by taking T to be the number of classes present in the problem $P = 1$ and the related penalty $\mathcal{P}_1 = \sum_{t=1}^{T} \sum_{t' \neq t} \Psi(K_t - K_{t'})$ we are considering the Joint Graphical Lasso problem (Danaher, Wang and Witten, 2014; Guo et al., 2011). Where ψ is the distance function among the precision matrices of the classes;
- by taking T as the number of time points in a time series, $P = 1$ and the related penalty $\mathcal{P}_1 = \sum_{t=1}^{T-1} \Psi(K_{t+1} - K_t)$ we are considering the Time-Varying Graphical Lasso. Here, again, the function Ψ is the temporal consistency function (Hallac, Leskovec and Boyd, 2015; Hallac et al., 2017a).

3.2 Performance Metrics for Graphical Models

The developing of a model is tricky mainly for the assessment of its performances in real-world contexts. Indeed, every model has some assumptions that may or may not be reflected in the data on which it is used. It is therefore necessary to pair its development with a quantitative and robust performance assessment strategy to prove its generalisation skills.
According to the learning task and to the experimental setting different performance metrics may be used. Nonetheless, the metrics should address two main questions:

(a) how much *likely* is a model for new (*i.e.*, unseen) data?

(b) Does the model represent the true structure of the system?

Question *(a)* could be address by observing the value of the likelihood as data changes. Indeed, it provides a direct indicator of the goodness of the model whilst not requiring the ground truth distribution by only the inferred model and on the data at hand. Therefore, it can be used also in the context of real-world data where the true underlying graph is not known. The likelihood, though, leads to over-fit and it is not suitable for distributions different from the Gaussian as it is impossible to compute. This topic will be further investigate later in Section 3.3.

Question *(b)*, on the contrary, requires the knowledge on the true structure to perform comparison. Such question can be addressed under two different perspectives depending on how we are interested in approximating the system, if we consider it:

- as a regression problem, we aim at approximating each edge weight;
- as a classification problem, we aim at approximating the structure.

3.2.1 *Metrics*

Since we are in the context of Markov Random Fields, the analysed graphs are undirected and, consequently, the adjacency matrix symmetrical. Let $\boldsymbol{K}$ be the true multiple graphical model, and $\hat{\boldsymbol{K}}$ its prediction, we consider their upper triangular part $y = \boldsymbol{K}^{(u)}$ and $\hat{y} = \hat{\boldsymbol{K}}^{(u)}$ respectively. These two vectors have dimension $L = TD(D-1)/2$.

STRUCTURE LEARNING AS REGRESSION If we consider the problem as a regression task we want to measure how good we are predicting the value of the edges. To this purpose we use a common metric called Mean Squared Error (MSE) that incorporates bias and variance of the model. This measure is scale-dependent and measures the distances between the entries of the inferred

precision matrix with respect to the entries in the true underlying precision matrix. It is defined as follows:

$$\text{MSE}(\hat{y}, y) = \frac{1}{L}\sum_{i=1}^{L}(\hat{y}_i - y_i)^2,$$

STRUCTURE LEARNING AS CLASSIFICATION The presence or absence of an edge in the graph, often, plays a more crucial role than its weight as it entails a connections (or its absence) between two nodes.Therefore, typically we want to assess whether we are able to detect the right edges by interpreting the edges as classes: class 1 is the existence of an edge, class 0 is its absence. We define *true/false positive* (TP/FP) to be the number of correctly/incorrectly existing inferred edges, while *true/false negative* (TN/FN) as the number of correctly/incorrectly missing inferred edges Hecker et al., 2009. The following classification metrics can be used for the assessment of the inference method:

- **Accuracy** consists in the percentage of correct predictions with respect to the total number. It ranges in the interval $[0, 1]$ with 1 being the maximum score and chance being the percentage of the most represented class over the total number of edges.

  $$\text{A}(\hat{y}, y) = \frac{1}{L}\sum_{i=1}^{L}\mathbb{1}(\hat{y}_i = y_i).$$

  The accuracy has the drawback of being inaccurate in the case of highly unbalanced classes.

- **Balanced Accuracy** is a simple extension taking into account the number of samples in each class. Balanced accuracy score (BA) ranges in the interval $[0, 1]$ and, for random classifier, is constrained to return 0.5 independently from the number of samples in each class.

  $$\text{BA}(\hat{y}, y) = \frac{1}{2}\cdot\left[\frac{\text{TP}}{\text{TP}+\text{TN}} + \frac{\text{TN}}{\text{TN}+\text{FP}}\right].$$

- **Precision** measures the positive predictive value as the fraction of positive samples over the total number of samples classified as positive:

  $$\text{P}(\hat{y}, y) = \frac{\text{TP}}{\text{TP}+\text{FP}}.$$

  It ranges in the interval $[0, 1]$.

- **Recall** also known as sensitivity or true positive rate, ranges in the interval $[0, 1]$ and measures the proportion of positive samples correctly classified as positive:

  $$\text{R}(\hat{y}, y) = \frac{\text{TP}}{\text{TP}+\text{FN}}.$$

- **F_1-score** The F_1-score is the harmonic mean of precision and recall, and it can be used to control both of them at the same time.

$$F_1(\hat{y}, y) = \frac{2TP}{2TP + FN + FP}.$$

- **Specificity** also known as *True Negative Rate*, measures the proportion of negative samples which are classified as negative, thus including false positive samples.

$$TNR(\hat{y}, y) = \frac{TN}{TN + FP}.$$

  It ranges in the interval $[0, 1]$.

- **False Positive Rate** also known as *fall-out*, it measures the proportion of negative samples which are incorrectly classified as positive, over all of the negative samples.

$$FPR(\hat{y}, y) = \frac{FP}{FP + TN}.$$

  It ranges in the interval $[0, 1]$.

- **Matthews Correlation Coefficient** provides a balances measure of the quality of a binary classifications and it can be used even if the classes are of very different sizes. It is the correlation coefficient between the observed and predicted binary classifications and returns a value in the interval $[-1, 1]$ where MCC$= 0$ corresponds to chance.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

OTHER STRUCTURAL SCORES Throughout the thesis we will encounter many synthetic experimental validations that will require additional measures aside the structural ones. In particular when dealing with latent data we will check whether the inferred marginalisation has a rank closed to the number of latent variables. Also, we will have a clustering procedure (Chapter 4) and we will validate the clustering accuracy using the V-measure.

- **Mean Rank Error** estimates the precision on the rank of an inferred matrix. Given an estimated matrix $\hat{\boldsymbol{L}} = (\hat{L}_1, \dots, \hat{L}_T)$ and the ground truth $L = (L_1, \dots, L_T)$ is defined as:

$$MRE = \frac{1}{T} \sum_{t=1}^{T} \left| \text{rank}(L_t) - \text{rank}(\hat{L}_t) \right|.$$

  A value close to 0 means that we are inferring the true number of latent variables over time, while, viceversa, a high value indicates a poor consideration of the contribution of the latent variables.

- **V-measure** is an entropy-based measure used to measure the goodness of a clustering algorithm by measuring how successfully the criteria of homogeneity and completeness have been satisfied. It ranges in an interval $[0, 1]$ where 0 is a bad clustering algorithm and 1 is a perfect one Rosenberg and Hirschberg, 2007

PRECISION-RECALL AND ROC CURVES Throughout this thesis we will often recur to the use of plots that summarise the goodness of the inferred graph. These graphs are the Receiver Operating characteristic Curve (ROC) and the Precision-Recall curve (PR).
Such curves depict the ability of a method (considered as a classification problem) to retrieve the truth as the weights of the edges varies. The ROC curve is created by plotting the true positive rate (TPR), or recall, against the false positive rate (FPR) at various thresholds of the edges weights and provides an intuition of the Type I error of the method. PR curves, instead, summarise the trade-off between the true positive rate (Precision) and the positive predictive value (Recall) at various thresholds of the edges weights.
Often, to obtain a number that gives us a general indicator of one curve we compute the Area Under the Curve (AUC) which is the integral of the area under the selected curve, an AUC equal to 0.5 means that the model performs as a random classifier, an AUC equal to 1 means that the model is perfectly accurate.

3.3 Multi-parameters Model Selection for Network Inference

Complex network inference methods have the major drawback of a high number of hyper-parameters that need to be tuned. Such problem is also known as *model selection* and it is one of the most challenging task in machine learning. Indeed, even if some theoretical bounds exist for network inference methods often they do not work in practice (Liu, Roeder and Wasserman, 2010). This is due to the fact the assumed sample size is typically not available as we put ourselves in high dimensional contexts in which $N \ll D$. The optimal models are therefore selected by empirically evaluating the performance on data. In the context of network inference this task is particularly difficult given the unsupervised nature of the problem, which therefore relies on likelihood scores (Bogdan, Ghosh and Doerge, 2004; Broman and Speed, 2002; Chen and Chen, 2008; Cheng, Shan and Kim, 2017; Danaher, Wang and Witten, 2014; Foygel and Drton, 2010; Hallac et al., 2017a; Siegmund, 2004) or stability measures (Liu, Roeder and Wasserman, 2010; Meinshausen and Bühlmann, 2010; Müller, Bonneau and Kurtz, 2016).
Likelihood and its penalisations (BIC (Guo et al., 2011) or AIC (Danaher, Wang and Witten, 2014)) are widely used in literature nested in a cross-validation schema. The best model is selected by taking the one that performs best in mean on multiple validation sets. The very first drawback is that this model selection strategy may lead to over-fit (Wasserman and Roeder, 2009). The second drawback is that likelihood-based scores may be conditionally applied based on the assumed probability distribution as the computation of the normalisation constant of the joint distribution may be infeasible — this is the case for Poisson, Ising, and Exponential graphical models (Allen and Liu, 2013; Ravikumar, Wainwright and Lafferty, 2010; Yang et al., 2013). On the

other hand, likelihood-based scores are easily extendible to the multi-network multi-hyper-parameters case. A valuable alternative are stability-based methods whose aim is to find the optimal value of the hyper-parameters that maximises stability of the inferred graph at multiple re-sampling of the data (Liu, Roeder and Wasserman, 2010; Meinshausen and Bühlmann, 2010). These criteria have proved, in the case of single network inference, to be more effective than likelihood-based scores (Liu, Roeder and Wasserman, 2010). Also, with assumptions of Poisson, Multinomial and other types of data they are the only possible choice. Such stability criteria were later extended to consider graphlets stability *i.e.* to verify the presence of non-isomorphic sub-graphs across experimental sub-sampling (Müller, Bonneau and Kurtz, 2016; Pržulj, Corneil and Jurisica, 2004).

In this section we provide a comprehensive description of the available likelihood-based scores for multi-parameters model selection; we then extend stability-based methods to the multi-parameters case, also including graphlets stability (Liu, Roeder and Wasserman, 2010; Meinshausen and Bühlmann, 2010; Müller, Bonneau and Kurtz, 2016).

3.3.1 *Likelihood scores for multi-parameters model selection*

Likelihood-based model selection methods rely on the possibility of computing the likelihood of the model under analysis. Therefore, as previously mentioned it is not possible to use the rest of the definition for some MRFs (*e.g.*, Ising, Poisson, Exponential (Allen and Liu, 2013; Yang et al., 2013, 2015)). This type of score can be used in a cross validation schema where, for each hyper-parameter combination, the model is trained on the learning set and the likelihood of the model is estimated on the independent test set. The hyper-parameters are selected based on the average maximum likelihood of the model across multiple splits of the data set.
When we have a set grid of hyper-parameters we can perform different cross-validation strategies:

- K-fold where data are partitioned in K folds, one is retained for validation and the remaining $K-1$ are used for train (see Figure 8).
- Monte Carlo Cross-Validation (MCCV) (Molinaro, Simon and Pfeiffer, 2005) that repeatedly splits the N samples of the data set in two mutually exclusive sets. For each split, $n \cdot (1/v)$ samples are labelled as *validation* set and the remaining $n \cdot (1 - 1/v)$ as learning set (see Figure 9).

When the hyper-parameters grid is not fixed we can either select the hyper-parameters at random in a specific interval (Bergstra and Bengio, 2012) or select them using Gaussian process-based Bayesian optimisation procedures (Snoek, Larochelle and Adams, 2012). The latter tends to reduce the computational times by choosing the best combination of hyper-parameters for each analysed data set, based on the Expected Improvement (EI) strategy.

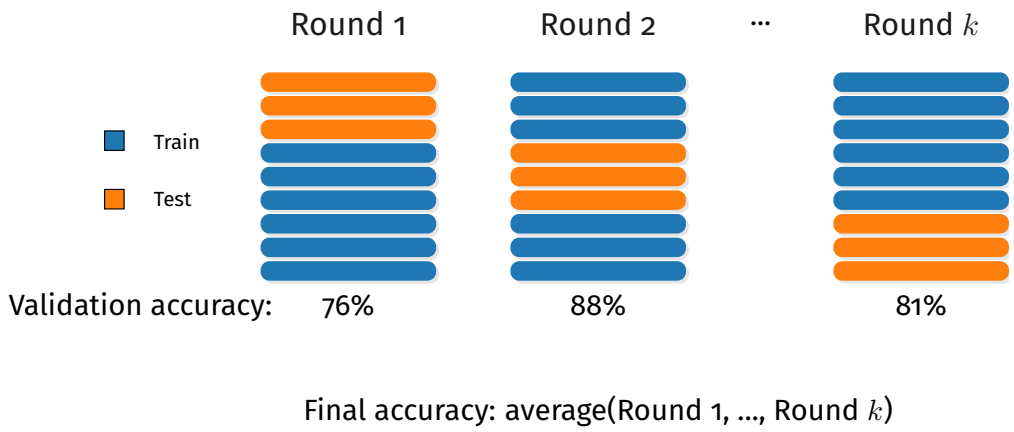


FIGURE 8. *Example of 3-Fold cross-validation schema. At each round the dataset is split in train and test with a fixed number of samples in each split. The train sets never overlap.*

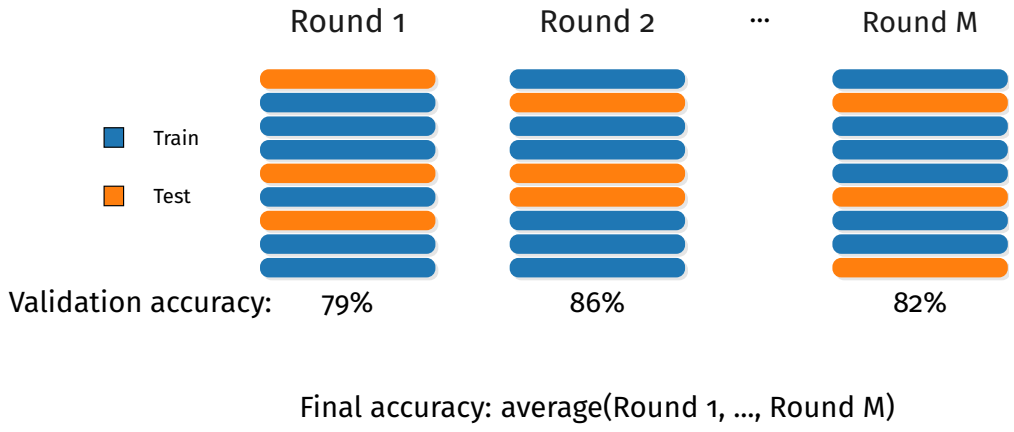


FIGURE 9. *Example of Monte Carlo cross-validation schema.At each round the dataset is split by randomly selecting a percentage of samples for training. Across rounds the train sets may overlap.*

Inside each cross-validation schema different scores can be used. Such scores, based on likelihood, are easily extendible to the multi-parameters multi-networks case as it suffices to take the mean of the scores on the single networks. Let us consider T graphs in D variables, for which we have $\boldsymbol{X} = (X_1, \dots, X_T)$ observations each of them having N_t samples. We denote Λ the generic hyper-parameters tuple of a model of the form in Equation (27) and the inferred adjacency matrices inferred with the specific choice of hyper-parameters are denoted as $\boldsymbol{K}_\Lambda$. Then, the generalised scores are:

GENERALIZED LIKELIHOOD SCORE It is the mean of the likelihoods computed on the single networks and it is defined as

$$\ell\ell_{GGM}(\boldsymbol{K}_\Lambda|\boldsymbol{X}) = \frac{1}{T}\sum_{k=1}^{T}\left[\frac{1}{N_t}\ell((K_\Lambda)_t|X_t)\right]$$

such score was used in Hallac et al., 2017a; Tomasi et al., 2018b to perform model selection on time-varying network inference.

GENERALISED BAYESIAN INFORMATION CRITERION(BIC) It penalises the likelihood by considering the *degrees of freedom* of the model in order to prevent overfitting for an increasing complexity of the model in analysis. In a graphical model selection problem the degree of freedom are the number of non-zero elements in the matrix (Stoica and Selen, 2004; Zou, Hastie and Tibshirani, 2007). Here, we take into account for the incremented number of degree of freedom given by the T graphs.

$$\text{BIC}(\boldsymbol{K}_\Lambda|\boldsymbol{X}) = \ell\ell_{GGM}(\boldsymbol{X}|\boldsymbol{K}_\Lambda) - \left(\sum_{t=1}^{T}\frac{\log(N_t)}{N_t}\right)\|\boldsymbol{K}_\Lambda\|_{od,0}$$

where $\|\boldsymbol{K}_\Lambda\|_{od,0}$ is the number of non-zero elements in the off-diagonal of the matrix $\boldsymbol{K}_\Lambda$. The BIC is a common score method for unsupervised problem as it leads to asymptotically consistent model selection when the number of variables D is fixed and the number of samples N_t increases. The AIC method for multi-networks (Danaher, Wang and Witten, 2014; Sakamoto, Ishiguro and Kitagawa, 1986) differs from this formulation only for the penalty that, instead of being proportional to the number of samples, is simply multiplied by 2. Due to this resemblance, we do not include it in the following comparison.

GENERALISED EXTENDED BIC (EBIC) It further penalises the likelihood with respect to BIC by adding a multiplicative part that depends on the number of variables

$$\text{EBIC}(\boldsymbol{K}_\Lambda|\boldsymbol{X},\epsilon) = \ell\ell_{GGM}(\boldsymbol{X}|\boldsymbol{K}_\Lambda) - \sum_{t=1}^{T}\left(\frac{\log(N_t)}{N_t} + 4\epsilon\frac{\log(D)}{N_t}\right)\|\boldsymbol{K}_\Lambda\|_{od,0}.$$

This score proposes a trade-off between the positive selection rate and the false discovery rate based on the choice of the positive parameters ϵ, which following the literature is selected as $\epsilon = 0.5$ (Bogdan, Ghosh and Doerge,

2004; Broman and Speed, 2002; Chen and Chen, 2008; Foygel and Drton, 2010; Siegmund, 2004). A similar extension suitable when analysing large graphs is defined as (Cheng, Shan and Kim, 2017).

$$\begin{aligned}\text{EBIC}_m(\boldsymbol{K}_\Lambda|\boldsymbol{X},\epsilon) = \ell\ell_{GGM}(\boldsymbol{X}|\boldsymbol{K}_\Lambda) - \\ \sum_{t=1}^{T}\left(\frac{\log(N_t)}{N_t} + 4\epsilon\frac{\log(TD(D-1)/2)}{N_t}\right)\|\boldsymbol{K}_\Lambda\|_{od,0}\end{aligned}$$

where $TD(D-1)/2$ is the total number of off-diagonal elements in the T precision matrices.

3.3.2 *Stability-based multi-parameters model selection*

Model selection approaches based on stability of the result are widely used in unsupervised settings as clustering (Lange et al., 2004; Von Luxburg, 2010). In the context of graphical models they were proposed in (Liu, Roeder and Wasserman, 2010; Meinshausen and Bühlmann, 2010). The two methods differ slightly, the first one propose the use of random sub-sampling to obtain more stable regularization paths while the second propose sub-sampling to directly detect the regularization parameter. Also, the first under-selects the edges while the second over-selects. The need of over-selecting the graph lies in the fact that it is computationally and manually easier to scan the detected edges for false positive rather than scanning all the missing edges looking for false negative. Indeed, the main assumption throughout this thesis is that the underlying graph of a system is sparse therefore the true number of edges between variables is low with respect to the total number of possible edges. Also, from an applicative perspective, it is better to falsely identify two variables that may have a connections rather than discarding a true connection between variables that may be important for the analysis of the system. The most used is called *Stability Approach to Regularisation Selection* (StARS) (Liu, Roeder and Wasserman, 2010), in which the best model is selected as the one that uses the minimum amount of regularisation still producing a sparse and stable graph under random sub-sampling of the initial dataset. StARS selects the best value for the hyper-parameter α by analysing the trend of stability as α varies. Indeed, as $\alpha \to \infty$ the inferred graph is completely sparse, *i.e.*, no edges are present. Therefore for $\alpha = \infty$ the graph is stable under random sub-sampling of the data. On the other hand, the same holds when $\alpha = 0$ as the graph is complete and therefore there is no variation in the inferred edges. Their approach selects the best α based on the possibility to order the regularisation parameters from the strongest regularisation to the weakest.

MULTI-PARAMETERS RELATION ORDER In the context of multi-parameters the ordering is trickier as different parameters act on different part of the inference which may or may not impact sparsity and stability. Here, we define a single parameter $\Lambda = (\alpha, \beta_1, \ldots, \beta_{P-th})$ as a tuple of hyper-parameters. Such hyper-parameters are not randomly positioned within the tuple but according

to their impact on the sparsity of the problem. Therefore, α that directly acts on the ℓ_1 penalty is the most important hyper-parameter, followed by a certain order of the βs such that β_I acts on edges stability more than $\beta_I I$ and so on and so forth. Note that in this case β_I does not necessarily denote β_1 but the one that has the highest impact on sparsity. When performing model selection, given the possible ranges for all the hyper-parameters and their order, we compute a grid of values naming each point of the grid as $\Lambda_i = (\alpha^i, \beta_I^i, \dots, \beta_{P-th}^i)$. We order the tuples Λ_i following the inverse lexicographic order so that α^i is the parameter that changes less frequently. In this way to the first tuple Λ_1 corresponds the most regularised (and therefore sparse) graph.

Example 1. We provide an example in the context of TVGL (Hallac et al., 2017a), here we have two hyper-parameters α and β, the former directly acts on the ℓ_1 penalty and it is therefore the most important while the latter acts on the temporal consistency. The generic tuple is defined as $\Lambda = (\alpha, \beta)$. Let us suppose that we take $\alpha \in \{0, 0.1, 0.5, 1\}$ and $\beta \in \{0.01, 0.1, 1, 10\}$ than all the possible combinations, in lexicographic order, are:

$$\Lambda_1 = (1, 10),\ \Lambda_2 = (1, 1),\ \Lambda_3 = (1, 0.1),\ \Lambda_4 = (1, 0.01),$$
$$\Lambda_5 = (0.1, 10),\ \Lambda_5 = (0.1, 1),\ \Lambda_6 = (0.1, 0.1),\ \Lambda_7 = (0.1, 0.01),$$
$$\dots$$
$$\Lambda_{14} = (0, 10),\ \Lambda_{14} = (0, 1),\ \Lambda_{15} = (0, 0.1),\ \Lambda_{16} = (0, 0.01),$$

The goal is to choose Λ^* such that the true graph E is contained in $E(\Lambda^*)$, i.e. the graph is over-selected.

SUB-SAMPLING Let $\boldsymbol{z} = (z_1, \dots, z_T)$ be the number of sub-samples drawn at random without replacement from each dataset X_t, such that each $z_t \in [1, N_t]$ is proportional to the original number of samples for each point t, *i.e.*, if $N_{t'} \geq N_t$ then $z_{t'} \geq z_t$. The suggested choice for z_t is

$$z_t = \min(10\sqrt{N_t}, 0.9N_t)$$

which allows to select a reasonable amount of sub-sample from the original dataset even when the original sample size is low (Liu, Roeder and Wasserman, 2010). Given the selected z_t there are possibly $M_t = \begin{pmatrix} N_t \\ z_t \end{pmatrix}$ sets of of possible sub-samples without repetitions. Ideally, one would sub-sample all the possible sub-sets,

$$M = \min(M_1, \dots, M_T),$$

but, for computational reasons, this is often un-feasible. We therefore opt to sub-sample a high number of times ($M \approx 100$) with the guarantee to reach the same stability results (Politis, Romano and Wolf, 1999).

3.3.2.1 *Single-edge stability computation - m-StARS*

Given the choice of $\boldsymbol{z}$ and M we end up with estimated edge matrices $E_m, m = 1, \dots, M$ for each Λ. Let $\boldsymbol{K}_\Lambda^m$ be the precision matrix obtained from the general

application of a graphical model solver ξ to the sub-sampled matrix $\boldsymbol{X}^m$. We want to obtain the probability that an edge is present across multiple repetitions. We approximate such value with a U-statistic of order M. Consider the binarised version of $\boldsymbol{K}_\Lambda^m$ denoted with $\bar{\boldsymbol{K}}_\Lambda^m$ we compute such approximation as

$$\hat{K}_\Lambda^z = \frac{1}{M}\sum_{m=1}^{M} \bar{\boldsymbol{K}}_\Lambda^m$$

Now we define

$$\hat{\xi}_\Lambda^z = 2\hat{K}_\Lambda^z(1-\hat{K}_\Lambda^z)$$

which is an estimate of twice the variance of the Bernoulli indicator of the edges of the matrices. It can be easily interpreted in the following way: for each pair of graphs obtained with the same parameter we compute how often they disagree on the existence of an edge. The value $\hat{\xi}_\Lambda^z \in [0, \frac{1}{2}]$ is the fraction of times they disagree. For each Λ its ξ measure the instability of the edges across sub-samples. This value is therefore compute for each possible edge (i,j) for $i,j = 1,\dots,D$ and each possible sub-graph t for $t = 1,\dots,T$. Then the parameter is selected as

$$\Lambda^* = \arg\min_{\Lambda} \left\{ \min \left[\frac{\sum_{t=1}^{T}\sum_{i<j}(\hat{\xi}_\Lambda^z)_{tij}}{T\begin{pmatrix} D \\ 2 \end{pmatrix}} \right] \leq \beta \right\} \tag{28}$$

where $\beta = 0.05$ is the significance level (Liu, Roeder and Wasserman, 2010). With this minimisation problem we take the highest monotonised values that is below the accepted threshold β. Note that, the result Λ^* depends on the block size z and therefore this method may have some efficiency loss in low dimension.

3.3.2.2 *Graphlets stability computation - mg-StARS*

StARS relies only on the single-edge stability which, by definition, ignores higher order stability relations. In Müller, Bonneau and Kurtz, 2016 they proposed an extension whose basic idea was to look for stability not only on the single edges but also for more complex topological structures known as *graphlets*(Pržulj, 2007; Pržulj, Corneil and Jurisica, 2004).

Definition 14 (Graphlet)**.** A graphlet is a small (typically 4 or 5 nodes) connected non-isomorphic sub-graph of a network.

Graphlets have to contain all the edges of the bigger networks between the nodes that are considered. They are widely used to characterize networks or to compare them. In particular, it is possible to count the number of time a certain type of graphlet appears in the graph obtaining a Graphlet Degree Vectors (Milenković and Pržulj, 2008; Pržulj, 2007). This vector can be used to

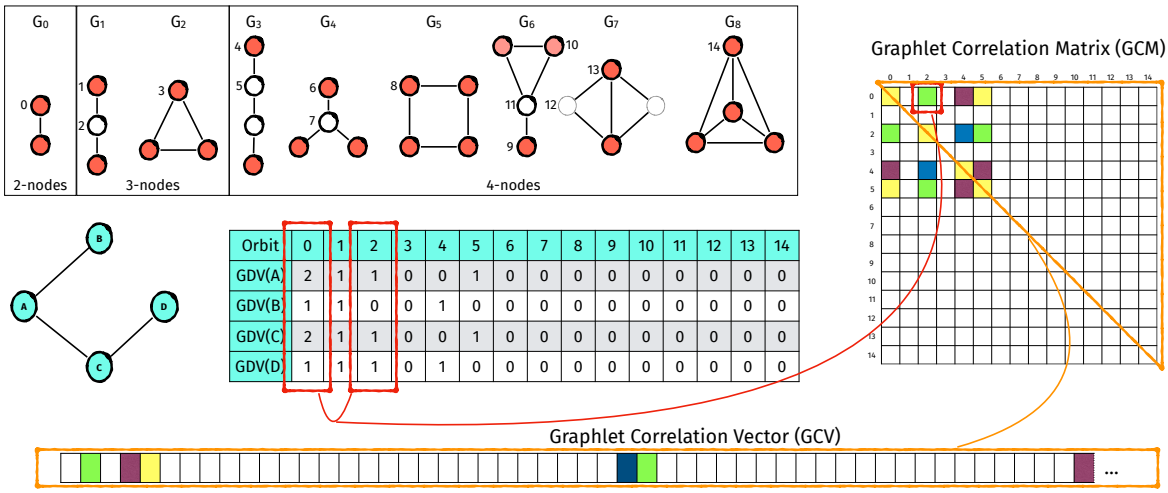


FIGURE 10. *Example of construction of a Graphlet Correlation Vector (GCV) used for the computation of stability of the inference method under multiple sub-sampling of the input data. On the top we have the representation of 4-nodes graphlets with the corresponding 14 orbits. Such graphlets are searched in the graph (blue nodes one on the left) and for each node we count how many times a specific type of orbit touches it. The, to compute the Graphlet Correlation Matrix (GCM) ($[0,1]^{14\times14}$) we compute the Spearman correlation coefficient of the obtained results. The ravelled upper triangular part of the GCM corresponds to the GDV of the graph.*

compute the Graphlet Correlation Matrix (GCM) (Sarajlić et al., 2016) where we can store for each vertex its graphlet degree vector and then compute the GCM between two graphs. The lower triangular of this matrix is the so-called Graphlet Correlation Vector (GCV) (Sarajlić et al., 2016) and can be used to compute distances between networks. Figure 10 shows a visual representation of the process necessary for the definition of a GCV.

Given the m-StARS approach we estimate $m = 1, \dots, M$ graphs with binarised adjacency matrix $\bar{\boldsymbol{K}}^m_\Lambda$ for each Λ. Each estimated graph has associated a GCV, here, since we have T graphs we would have a tuple of GCVs vectors $\boldsymbol{\rho}^m_\Lambda = ((\boldsymbol{\rho}^m_\Lambda)_1, \dots, (\boldsymbol{\rho}^m_\Lambda)_T)$. Then the graphlet variability (or *instability*) for fixed Λ over M estimates is defined as the avarage Euclidian distance among all GCVs:

$$\hat{d}^z_\Lambda = \frac{2}{TM(M-1)} \sum_{t=1}^{T} \sum_{m'>m} \|(\boldsymbol{\rho}^m_\Lambda)_t - (\boldsymbol{\rho}^{m'}_\Lambda)_t\|_2.$$

Such measure goes to zero for very sparse and very dense graphs, but differently from the measure used in StARS it is highly variable and therefore cannot be monotonised. Nonetheless, such measure can be used to support m-StARS by requiring simultaneously edge and graphlet stability.

This is achieved by selecting the Λ in an interval that is detected by looking at single edge stability, in particular we determine the best hyper-parameters tuple as

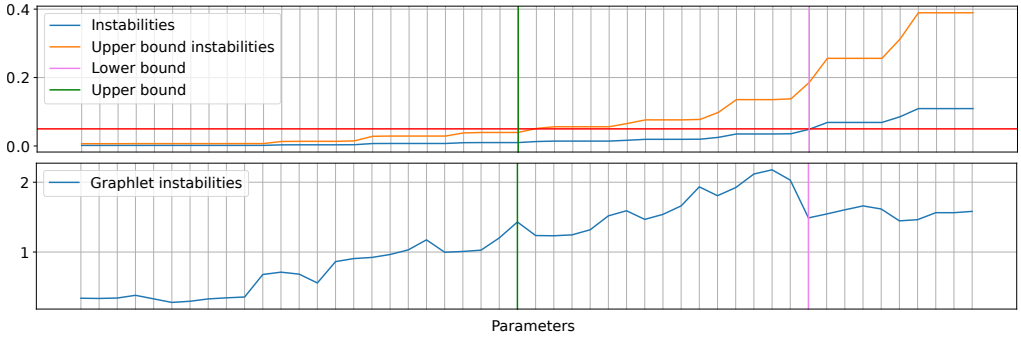


FIGURE 11. *Instabilities curves obtained applying the stability-based model selection methods for multiple parameters (m/mg-StARS) on the Joint Graphical Lasso. On the top we have the single edge instabilities (blue line) and their upper bound (orange line), The vertical lines delimit the space in which to look for graphlet stability (bottom plot line).*

$$\Lambda^* = \underset{\Lambda \in [\Lambda^{lb}, \Lambda^{ub}]}{\operatorname{argmin}} \hat{d}^z_\Lambda$$

where the value Λ^{lb} is selected as in Equation (28) and Λ^{ub} is selected as

$$\Lambda^{ub} = \arg\min_{\Lambda} \left\{ \min \left[\frac{\left(\sum_{t=1}^{T} \sum_{i<j} 4(\hat{\xi}^z_\Lambda)_{tij}(1 - (\hat{\xi}^z_\Lambda)_{tij}) \right)}{T \begin{pmatrix} D \\ 2 \end{pmatrix}} \right] \leq \beta \right\}$$

note that in this way we are defining an upper-bound curve of the single-edge instability curve (this can be visually observed in Figure 11, where in the top panel the orange curve lies above the blue curve).

Proposition 2. The value $\sum_{t=1}^{T} \sum_{i<j} 4(\hat{\xi}^z_\Lambda)_{tij}(1 - (\hat{\xi}^z_\Lambda)_{tij})$ is an upper-bound value of $\sum_{t=1}^{T} \sum_{i<j} (\hat{\xi}^z_\Lambda)_{tij}$.

Proof. This can be proved easily as

$$2 \frac{4}{\begin{pmatrix} D \\ 2 \end{pmatrix}} \sum_{t=1}^{T} \sum_{i<j} \hat{\xi}^z_\Lambda \left(1 - \frac{1}{T \begin{pmatrix} D \\ 2 \end{pmatrix}} \sum_{t=1}^{T} \sum_{i<j} \hat{\xi}^z_\Lambda \right) \geq \frac{2}{\begin{pmatrix} D \\ 2 \end{pmatrix}} \sum_{t=1}^{T} \sum_{i<j} \hat{\xi}^z_\Lambda$$

is true if and only if

$$\frac{1}{\begin{pmatrix} D \\ 2 \end{pmatrix}} \sum_{t=1}^{T} \sum_{i<j} \hat{\xi}^z_\Lambda \leq \frac{1}{2}$$

that is trivially true given that $\hat{\xi}^z_\Lambda \in [0, \frac{1}{2}]$ (Liu, Roeder and Wasserman, 2010). □

3.3.3 *Synthetic data experiments*

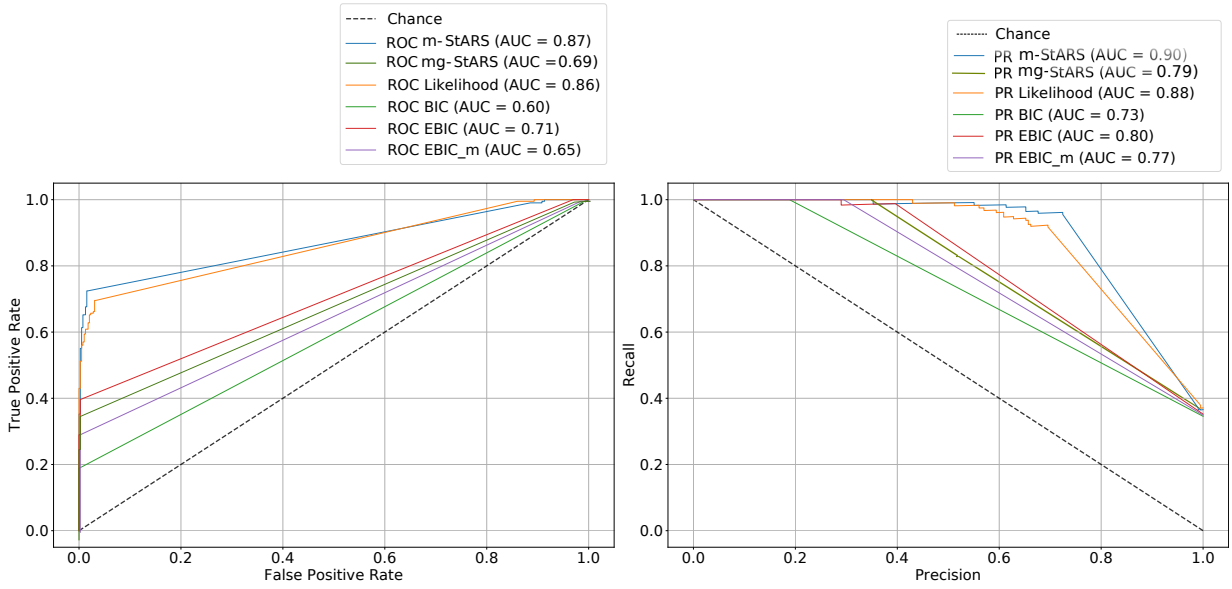
We designed four experiments to assess the efficacy of the proposed model selection method. We tested m-StARS and mg-StARS against likelihood-based model selection schema. We also included a comparison with the single parameter model selection StARS, when possible. For the model selection with likelihood-based scores we used a 3-fold cross-validation schema training the model on a subset of data and testing it on the remaining part. We tested all model selection strategies on three GGMs model with multiple hyper-parameters, in particular the Joint Graphical Lasso (JGL) (Danaher, Wang and Witten, 2014), the Time-varying Graphical Lasso (TGL) (Hallac et al., 2017a) and the Latent Graphical Lasso (LGL) (Chandrasekaran, Parrilo and Willsky, 2010) . We generated data in the following way:

- for JGL experiment we generated a random graph of 20 nodes each of the having degree 3 using the Networkx package (Hagberg, Swart and S Chult, 2008). This randomly generate graph represent the set of edges that is shared across all the classes, that we chose to be three ($T = 3$). In order to generate the graphs of the single classes we randomly add some edges.

- for TGL, we devised two experiments in which we generated 10 time-evolving networks of $D = 100$ variables ($T = 10$) with two different evolution schema: smooth changes (TGL-ℓ_2) and punctual changes (TGL-ℓ_1). In this specific case we compared also with the model selection performed on single parameters, *i.e.*, we considered each of the 10 networks separately and used the Graphical Lasso. We called such experiments (GL-ℓ_1) and (GL-ℓ_2).

- for LGL, we generated a perturbed observed network on 100 observed variables with 5 latent (therefore we do not have multi-networks inference and $T = 1$). For the generation of the data that satisfies the theoretical constraints (see Section 2.4) we followed the generation schema presented in (Yuan, 2012).

Both JGL and TGL have two hyper-parameters α that regulates sparsity and β that regulates the similarity of the network across classes/times, we sorted them as $\Lambda = (\alpha, \beta)$. LGL has two hyper-parameters α that regulates sparsity and τ that controls the amount of estimated latent variables, we ordered them as $\Lambda = (\alpha, \tau)$. For all the experiments and all the classes/times we generated $N = 100$ samples. We adapted the range of parameters to the specific case and we used 4-nodes graphlets in the mg-StARS computation. For all experiments we computed Precision-Recall (PR) and ROC curves by considering the edges of the graphs as binary classes.



(A) *ROC curves* (B) *Precision-Recall curves*

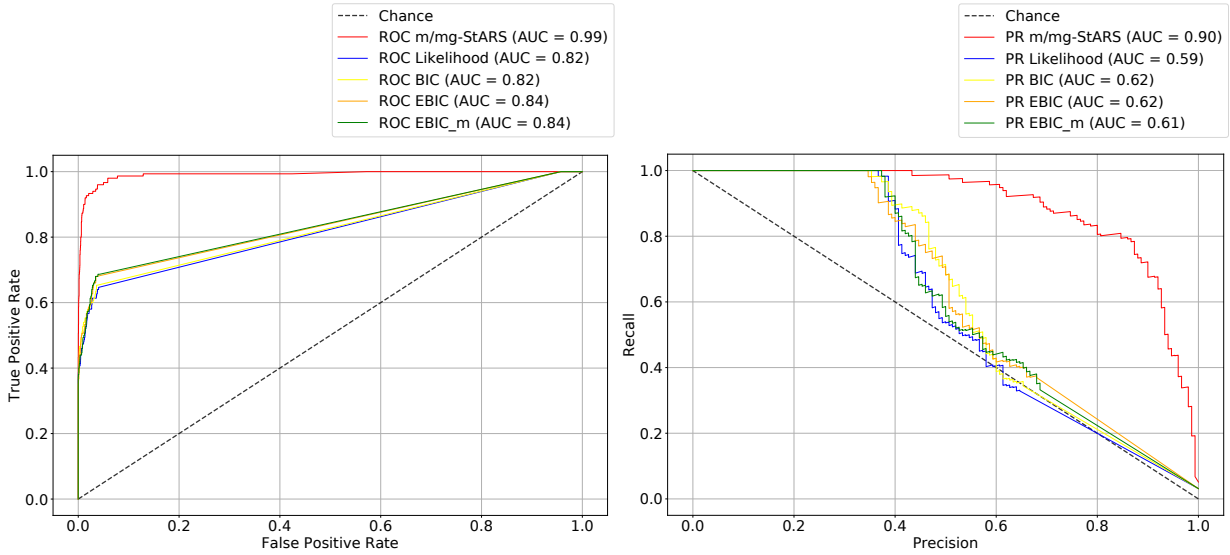
FIGURE 12. *Results comparison in terms of ROC and PR curves of the performance of the Joint Graphical Lasso for different hyper-parameters (model) selection methods as m/mg-StARS, likelihood, BIC, EBIC and EBIC_m.*

3.3.4 *Results*

Stability-based methods for model selection provide better results in terms of accuracy in structure recovery than likelihood-based strategies.

In Figure 11 an example of instabilities obtained applying our method for the experiment on JGL. It is noticeable that the instabilities assume a sort of step ascent, which assesses the validity of ordering the hyper-parameters according to their impact to sparsity. We can observe that in this case the model selected with m-StARS or with mg-StARS is different. In the other experiments, that we do not report, both algorithms selected the same model. The differences among the performances of m-StARS, mg-StARS, and likelihood-based scores is reported in the of Figure 12. Looking at the curves we observe that the model selected for mg-StARS performs worse than likelihood-based scores, while, if we simply use m-StARS we obtain better results.

In Figure 13 we report the results obtained for the experiments on LGL, which is the only case in which we have multiple hyper-parameters but only one inferred network. Again m/mg-StARS perform better than likelihood-based scores. In this case it is particularly evident as we generated synthetic data from a spurious precision matrix. Therefore likelihood scores tend to overfit the dataset while m-StARS, that seeks for a stable result, obtain higher scores. Lastly, in Figures 14, 15, 16, 17 we show the curves obtained for TGL-ℓ_1 and TGL-ℓ_2 in the single network (14, 16) and multiple network (15, 17) cases. Note that, m-StARS (which is equivalent to mg-StARS) is the one providing the best performance, both in single and in multiple network case. It is also



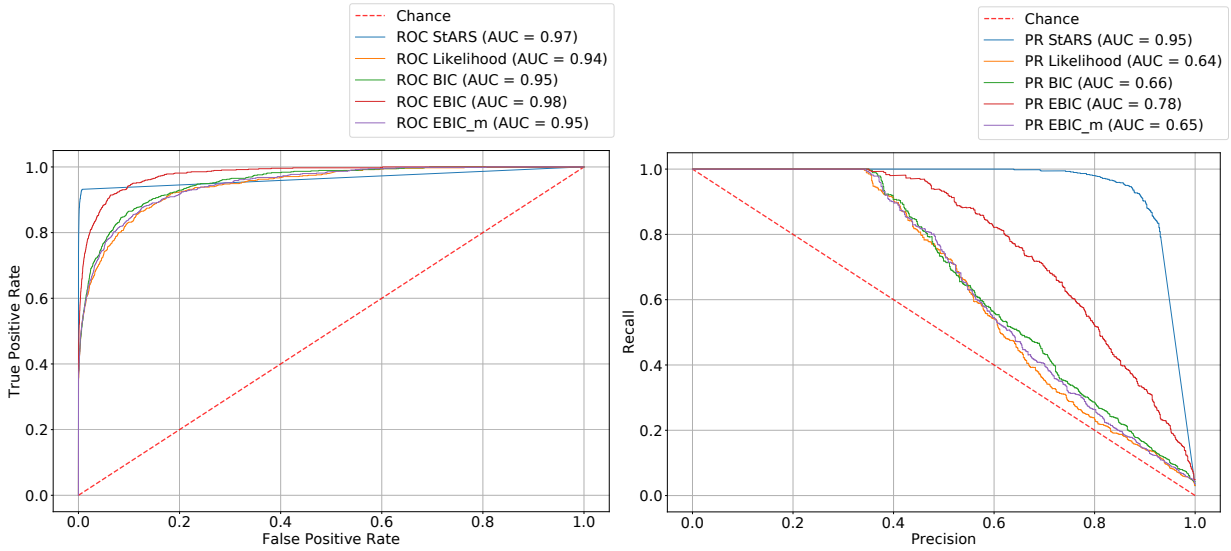
(A) *ROC curves* (B) *Precision-Recall curves*

FIGURE 13. *Results comparison in terms of ROC and PR curves of the performance of the Latent Graphical Lasso for different hyper-parameters (model) selection methods as m/mg-StARS, likelihood, BIC, EBIC and EBIC_m.*

worth mentioning that, when considering the 10 networks as single independent networks, GL-ℓ_1 and GL-ℓ_2 experiments, the results are always notably worse with respect to considering them as a time evolving network. Lastly, we wanted to point out that there is no consistent difference between likelihood and its penalisation when inserted in a cross-validation schema. None of these scores outperforms the others.

3.4 Summary

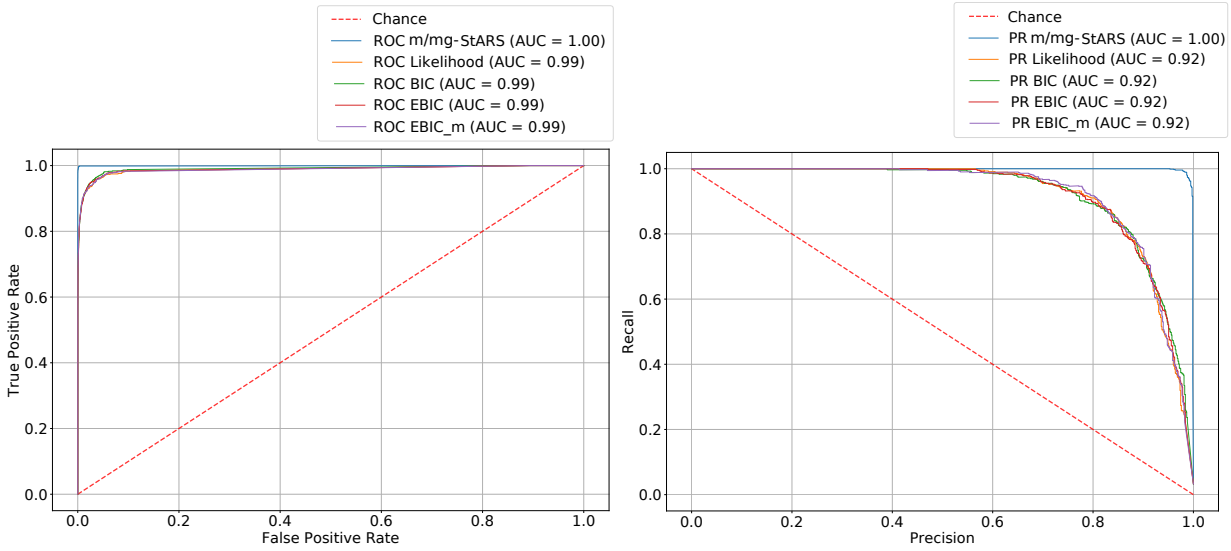
In this chapter we presented the measured to assess the goodness of an inferred graphs when the ground truth is known and we presented an extension for model selection based on stability of the result for network inference methods that present more than one hyper-parameter. We showed the validity of the proposed stability-based criterion on Gaussian Graphical Models comparing m-StARS and mg-StARS with likelihood-based cross validation schema noticing that m-StARS always provides a better estimate of the model. We remark that, in cases of non-Gaussian data, stability-based model selection criteria are the only possible choice. Therefore, a suitable method for multi hyper-parameters selection is necessary for further exploring more complex models on other distributions (Lee and Hastie, 2015; Yang et al., 2014; Žitnik and Zupan, 2015). From the methodological perspective graphlets stability has proved to be less effective than single edge stability. For future work it would be interesting to exploit other types of stability possibly looking at topological descriptors capturing higher order relations such as persistent homology (Bergomi et al., 2019). Also, it could be of interested to further validate the



(A) *ROC curves* (B) *Precision-Recall curves*

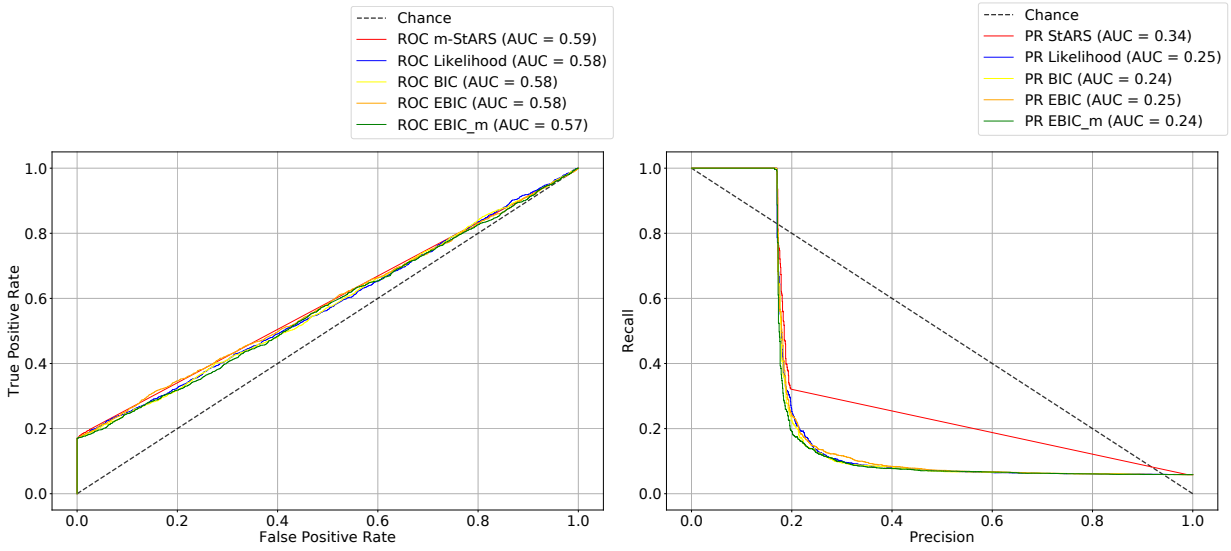
FIGURE 14. *Results comparison in terms of ROC and PR curves of the performance of the Time-Varying Graphical Lasso model (with temporal evolving behaviour ℓ_1) considered as single separated networks for different hyper-parameters (model) selection methods as m/mg-StARS, likelihood, BIC, EBIC and EBIC_m.*

proposed stability-based method to check for other ordering of the tuples Λ and observe their empirical performance.



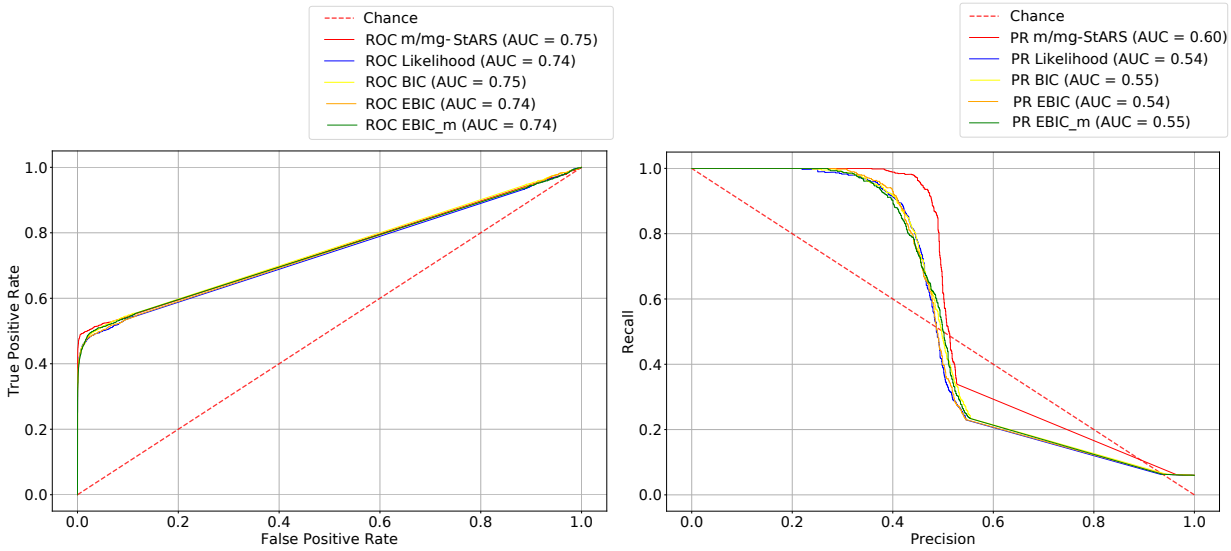
(A) *ROC curves* (B) *Precision-Recall curves*

FIGURE 15. *Results comparison in terms of ROC and PR curves of the performance of the Time-Varying Graphical Lasso model (with temporal evolving behaviour ℓ_1) for different hyper-parameters (model) selection methods as m/mg-StARS, likelihood, BIC, EBIC and EBIC_m.*



(A) *ROC curves* (B) *Precision-Recall curves*

FIGURE 16. *Results comparison in terms of ROC and PR curves of the performance of the Time-Varying Graphical Lasso model (with temporal evolving behaviour ℓ_2) considered as single separated networks for different hyper-parameters (model) selection methods as m/mg-StARS, likelihood, BIC, EBIC and EBIC_m.*



(A) *ROC curves* (B) *Precision-Recall curves*

FIGURE 17. *Results comparison in terms of ROC and PR curves of the performance of the Time-Varying Graphical Lasso model (with temporal evolving behaviour ℓ_2) for different hyper-parameters (model) selection methods as m/mg-StARS, likelihood, BIC, EBIC and EBIC_m.*

4

Methods for Generalised Temporal Network Inference

Part of this chapter content is present in the following publications:
Federico Tomasi, Veronica Tozzo, and Annalisa Barla. Temporal Patterns Detection in Time-Varying Graphical Models. Submitted. (2019)

The main focus of this thesis is to study and extend methods for the inference of dynamic graphical models, able to describe a system as it changes. Dynamic graphical models consider the change of the system behaviour as a change in the structure of the network itself. We argue that the use of such models is fundamental in the analysis of real-world systems as they are typically composed of many entities (that we mathematically model with variables) which may behave and interact differently in time. The inference can be guided by a certain consistency in how these entities behave in close time points, a concept that we identified as *temporal consistency* (see Section 1.7) Such concept has been exploited only in the case of Gaussian Graphical Models (GGMs) (Bianco-Martinez et al., 2016; Hallac et al., 2017a; Harutyunyan et al., 2019) but not for other type of probability distributions. We argue that it would be necessary to extend it to other distribution as real-worlds observations may be of different type, *e.g.*, continuous, binary or counts (Yang et al., 2015). Also, temporal consistency used in GGMs (Hallac et al., 2017a) only considers *Markovianity*, *i.e.*, each time point exclusively depends on the previous one. Markovianity may be a shortcoming when the analysed phenomenon presents long term or recurrent relationships among time points. Here, we propose a general model for network inference that can be instantiated in principle with any log-likelihood allowing for the analysis of different data types, for considering possibly non-Markovian relationships among time points. We also show how, in case of no knowledge on the type of temporal dependency, it can be automatically inferred. Lastly, we show how one could instantiate our general model also to consider multi-class problems which, again, are present in literature only for GGMs (Danaher, Wang and Witten, 2014).

OUTLINE The rest of this Chapter is organised as follows. Section 4.1 introduces the problem of inferring a dynamical graphical models under different assumptions. Section 4.2 illustrates a clustering approach that can be coupled with such model for the automatic inference when the temporal dependencies are not known a priori. Section 4.3, 4.4 and 4.5 present the three instantiations of the model in case of Gaussian, Ising or Poisson models with the related synthetic data experiments. Section 4.6 shows the use of our general model for the inference of multi-class graphs. Section 4.7 concludes with a brief discussion on our contribution and future research directions.

4.1 Temporal Consistency and Dependency

The concept of temporal consistency that we introduced in Section 1.7.1 is deeply connected to how we considered time points to be related. Indeed, our definition of consistency (Definition 11) states that two points are consistent if their distance (defined by a function Ψ) is small. In Hallac et al., 2017a they defined temporal consistency by assuming Markovianity. This implies that one point is *dependent* only to the previous one and, therefore, their structure is *consistent*, *i.e.*, given two points t and $t+1$ they have a small distance $\Psi(K_{t+1} - K_t)$.
Nonetheless, reality often presents more complex dependencies than Markovianity. Consider as an example weather data, that presents highly seasonal, weekly and daily recurrence of conditions. In this case, the network structure of variables in a one hour time span would be dependent not only on the previous hour but also on the previous day at the same time, the previous week and same season the year before.
Therefore, we couple the concept of consistency with the one of *temporal dependency*. Consistency is a function of two given networks that provides a measure of how much they are similar in structure. Dependency is a function of time that provides temporal instants in which the phenomenon under analysis presents similarly to the current time point.

Definition 15 (Dependency)**.** Two time points are said *dependent* if they model the same behaviour of the system under observation.

Thereby, if the phenomenon at time point t is dependent on the phenomenon at time point t', this implies that the vice-versa holds (t' is dependent on t). As we assume network structure and the evolution of the system to be bi-univocally linked, temporal consistency and dependency translate in a similarity of the network structures of dependent time points.
Our goal is to provide a general model that allows to consider any probability distribution under possibly non-Markovian temporal dependencies. In Figure 18 we provide an example of possible situations we aim at modelling. If we consider time t_4, the red rectangles in each row define the networks to which t_4 is dependent from. Their height indicates how much t_4 should depend (or be similar) to the other networks where the similarity is defined by the consistency function Ψ.

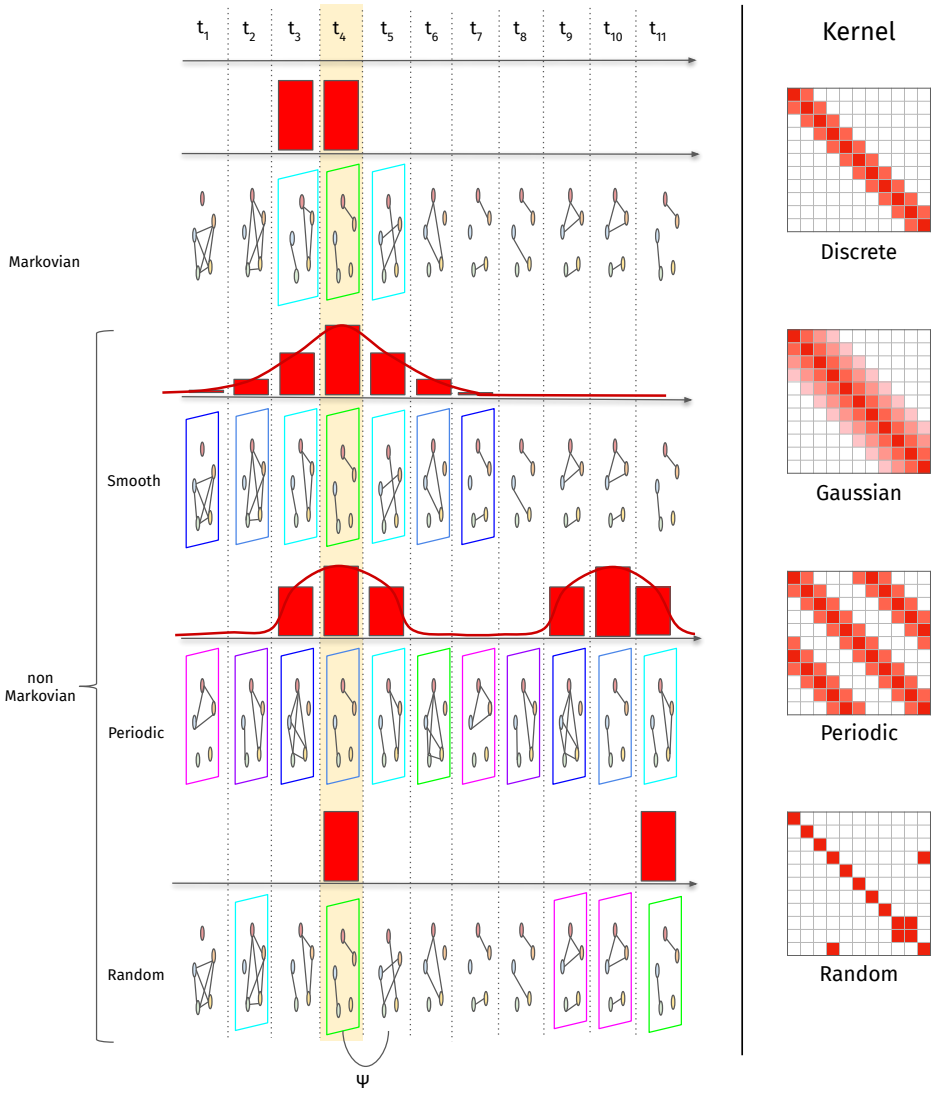


FIGURE 18. *Examples of temporal evolving networks with Markovian (first row) and non-Markovian temporal dependencies (smooth, periodic and random). If we focus on time t_4 (highlighted in yellow) we observe on top of the network structure red rectangles that identify the time points from which t_4 is dependent on. Such dependency is forced on the structure by a consistency function Ψ. On the right side the related kernels that allow for the imposition of specific temporal dependency patterns,* i.e., *they provide a more structured representation of the rectangles.*

If we know *a priori* the type of dependency of data we are analysing, such knowledge can be easily embed in any temporal network inference method through a stationary kernel κ that, at each time point, tells us which are the dependent others (see the right side of Figure 18). Therefore, by decoupling consistency type (Ψ) and dependency type (κ) we are able to model higher-order

temporal relations such as smooth changes (2^{nd} row of Figure 18) and periodicity (3^{rd} row of Figure 18). Note that, for type of dependencies as smooth or periodic, we can easily recur to a mathematical definition of the kernel. While for cases like the random one (4^{th} row of Figure 18) we may need to manually construct the kernel. An example of this case is a neurological study in which we provide various stimuli to a subject at random instant in time
Temporal consistency and dependency could improve the inference of dynamic networks when the amount of samples available per each time point is small, as information from distant dependent time points is exploited to retrieve the true network structure.

4.1.1 *Model*

We consider a dynamic undirected graph $G = [V, E_t]$, where $V = \{1, \dots, D\}$ is a finite set of nodes that represent variables, and $E_t \subseteq V \times V$ is the set of edges between the nodes at a particular time t, for $t = 1, \dots, T$. We define a *dynamic graphical model* as a non-stationary probability distribution p_G belonging to the exponential family on $X_1, \dots, X_D$ that factorises according to the graph G.
The conditional independence between two variables X_i and X_j given all the others at time t is encoded in E_t, in particular they are independent if $(i, j) \notin E_t$. Such conditional independence can also be encoded in an adjacency matrix K that contains the structure of G in such a way that $K_t(i, j) = 0$ if and only if $(i, j) \notin E_t$ (Lauritzen, 1996).
At each time point t, consider a set of N_t observations $X_t \in \mathcal{X}^{N_t \times D}$ where each sample is a D-dimensional vector drawn from a multivariate distribution p_G whose sample space is the set $\mathcal{X}$. In particular, as explained in Section 1.4, 1.5 and 1.6 we have that $\mathcal{X} = \mathbb{R}$, $\mathcal{X} = \{-1, 1\}$, $\mathcal{X} = \mathbb{N}$ for the Gaussian, Bernoulli and Poisson distribution assumptions respectively.
We want to define a general network inference method that, from such observations, learns a series of adjacency matrices $\boldsymbol{K} = (K_1, \dots, K_T)$. Note that the number T is a choice that we need to do a priori, indeed each network at time t is considered as a discretisation of the time-series in time points. Therefore, given multi-variate time series of length τ we split them in chunks of equal temporal span in such a way that each chunk has length $N_t = \frac{\tau}{T}$. We consider each observations in a chunk to be i.i.d. (Hallac et al., 2017a).
The inference is thus guided by temporal consistency with possibly non-Markovian dependencies. In order to define which are the dependent time points we introduce in the model a stationary *kernel*.

Definition 16. A kernel $\kappa \in \mathcal{S}_+^T$ is a positive semi-definite matrix that encodes, at each entry $\kappa(t, t')$, the strength of dependency between the adjacency matrices at time t and t'.

Such kernel encodes the strength of how much two network at different time points should be similar in such a way that, samples belonging to different (but related) time points, can drive the inference toward a better estimation of the structure.

In order to impose possibly non-Markovian temporal dependencies we add a penalty $P_{\Psi,\kappa}(\boldsymbol{K})$. Such penalty depends on the kernel κ that encodes the type of dependency and the consistency function Ψ that defines the type of similarity between dependent graphs (possible choice for Ψ are presented in Section 1.7) and it is defined as

$$P_{\Psi,\kappa}(\boldsymbol{K}) = \sum_{s>t}^{T} \kappa_{st} \Psi(K_s - K_t) = \sum_{t=1}^{T-1} \sum_{t'=1}^{T-t} \kappa_{tt'} \Psi(K_{t+t'} - K_{t'}). \tag{29}$$

Such penalty is applied to a joint inference of T graphical model selection problems as follows

$$\underset{\boldsymbol{K}\in\mathcal{S}^T}{\text{minimize}} \sum_{t=1}^{T} \left[- N_t \ell(X_t|K_t) + \alpha \|K_t\|_{\text{od},1} \right] + P_{\Psi,\kappa}(\boldsymbol{K}) \tag{30}$$

where $\ell(K_t|X_t)$ is the log-likelihood to be instantiated, different choices of ℓ lead to different inferred networks. The penalty $\|\cdot\|_{od,1}$ is the off-diagonal ℓ_1-norm, which promotes sparsity in the adjacency matrices (excluding the diagonal). The constraint $K_t \in \mathcal{S}^T$ forces the adjacency matrices to be symmetric. Note that the functional in Equation (30) has the same form of Equation (27) in Section 3.1 thus allowing for the use of all the model selection strategies previously introduced (Chapter 3).

4.1.2 *Stationary Kernels*

We exploit stationary kernels to model the pair-wise similarities dependency strength as they are defined based only on the distance between two points and not on their identity. A strength equal to zero ($\kappa[t,t'] = 0$) implies that time t and t' are independent from each other and, therefore, no consistency is forced on them during the inference.
A particular kernel reflects the prior information on the behaviour of the system. Here, we introduce three kernels that will be used in this thesis: the discrete, Gaussian and periodic kernel (see right side of Figure 18 for a visual representation of the three kernels with $T = 11$). For a comprehensive overview of possible kernel functions see Rasmussen, 2003, Chapter 4.
Consider now two time points t_i and t_j, and a generic distance d.

DISCRETE A kernel which enforces similarity only on consecutive points can be defined as

$$\kappa(t_i,t_j) = \begin{cases} 1, & \text{if } t_i = t_j \\ \beta, & \text{if } t_i = t_{j+1} \text{ or } t_i = t_{j-1} \\ 0, & \text{otherwise} \end{cases} \tag{31}$$

where $\beta > 0$ measures the strength of how similar t_i and t_j are. This kernel assumes all points to be equally distant and the sequence to be Markovian, that

is two points are related if consecutive, otherwise are deemed as independent. Note that using this kernel with Gaussian data assumption leads to the TGL model (Hallac et al., 2017a).

PERIODIC Periodic trends, appearing at regular intervals over a time series, may be captured by using a periodic kernel. A common choice for this type of kernel is the exponential-sine-squared (ESS), of the form

$$\kappa_{ESS}(t_i, t_j) = \exp\left(-\frac{2}{s^2} \sin^2\left(\frac{\pi d(t_i, t_j)}{p} \right) \right),$$

where p is the periodicity and s the length scale of the kernel.

RADIAL-BASIS FUNCTION Patterns that decay slow, highly influencing their neighbouring time points and less the further time points, may be captured by using a radial-basis function (RBF) kernel of the form

$$\kappa_{RBF}(t_i, t_j) = \exp\left(-\frac{d(t_i, t_j)^2}{2s^2} \right),$$

where s the length scale of the kernel.

4.1.3 *Minimisation Algorithm*

We consider three different exponential class sub-families for the instantiation of the likelihood ℓ: the Gaussian, the Bernoulli (Ising) and the Poisson. We call the related inference methods, which depend from a particular kernel κ, *Kernel Temporal Graphical Lasso* (TGL$_\kappa$), *Kernel Temporal Ising Graphical Model* (TIGM$_\kappa$), and *Kernel Temporal Poisson Graphical Model* (PIGM$_\kappa$) respectively. Note that the name of the first method, TGL$_\kappa$, slightly differs from the other two to be consistent with the naming used in the state-of-the-art for Gaussian based network inference methods.
The minimisation algorithm of these model is based on the Alternating Directions Method of Multiplier (ADMM), an optimisation method that divides the problem into sub-problems. Such division makes the optimisation algorithm easily customisable for any distribution assumption, indeed the likelihood enters only in one of the step necessary for the minimisation. Furthermore, given the form of the functional the algorithm is guaranteed to converge to a global optimum of the problem (Boyd et al., 2011). For readability, we put all the mathematical derivations in Appendix A.

4.2 Automatic Inference of Temporal Dependencies

In the previous section we made an important assumption: the dependency pattern is known a priori. Nevertheless, this assumption is not always satisfied. Hence, the imposition of a kernel in the model is not feasible.

Algorithm 3 Automatic inference of non-Markovian dependencies

Inputs: s (length scale), β strength, k number of network clusters.
$\kappa^0 = \text{RBF}_\beta$
for $\iota = 1, \dots$ **do**
  $\boldsymbol{K}^\iota = \text{TGL/TIGM/TPGM}_{\kappa^{\iota-1}}$
  compute S^ι as from Equation (32)
  clusters$^\iota$ = AgglomerativeClustering(S^ι, k)
  compute C^ι as from Equation (33)
  $\kappa^\iota = \kappa^0 + \beta C^\iota$
  if $C^\iota == C^{\iota-1}$ **then**
    break

To overcome this issue we propose to couple the previous TGL_κ, TIGM_κ and PIGM_κ with a clustering procedure that automatically detects the most similar (and therefore dependent) networks in time. Following the same naming criteria we call such methods *Temporal Graphical Lasso with Pattern detection*(TGL_P), *Temporal Ising Graphical Model with Pattern detection* (TIGM_P), and *Temporal Poisson Graphical Model with Pattern detection* (PIGM_S).
We simultaneously infer both the clusters and the networks. We do not impose, to networks belonging to the same cluster, the exact same structure. Related approaches can be found in literature. In the context of GGMs, starting from a dynamical network, Ho, Song and Xing, 2011 cluster the single networks in time. Hallac et al., 2017b jointly estimate a dynamical network and patterns of network similarity. This approach (TICC) is the most similar to our automatic inference but, differently from us, imposes for each cluster the same network structure, which may be limiting in real cases where the structure of the dynamical network may be similar but not necessarily identical at different time points.

The coupling with a clustering algorithm introduces two problems: we need a further hyper-parameter k, *i.e.*, the number of clusters; the functional has now two unknowns (κ and $\boldsymbol{K}$) which multiply each other and make the problem non-convex.

4.2.1 *Minimisation Algorithm*

The minimisation of the problem with the two unknowns is performed with an alternating minimisation procedure. We fix the inferred networks to find the clusters and then, given the clusters, we improve network inference. This is repeated until convergence. Consider the model in Equation (30) with fixed Ψ and unknown kernel κ. We could explore the initial similarities between time points by using an initial kernel that we will call κ^0 which impose only a temporal similarity through an *RBF* kernel (also a discrete would be a suitable choice). The resulting network inferred at iteration ι is used to compute a sim-

ilarity matrix S that depends on the function Ψ. In particular, at each iteration ι, we compute

$$S^{\iota}[t,t'] = 1 - \frac{\Psi(K_t - K_{t'}) - \min\limits_{m,m'=1,\dots,T} \Psi(K_m - K_{m'})}{\max\limits_{m,m'=1,\dots,T} \Psi(K_m - K_{m'})} \tag{32}$$

Each entry of the matrix S^{ι} has a value in the interval $[0,1]$ where 1 means that the network at time t and t' are identical and 0 corresponds to the most dissimilar networks. The matrix S^{ι} is then used as input for a Hierarchical Clustering algorithm (Defays, 1977) which provides in output k clusters. Of these clusters we build the connectivity matrix C^{ι} as follows:

$$C^{\iota}[t,t'] = \begin{cases} 1, & \text{if t' and t belong to the same cluster} \\ 0, & \text{otherwise} \end{cases} \tag{33}$$

which is a symmetric matrix with diagonal 1. The final kernel is thus defined as

$$\kappa^{\iota} = \kappa^0 + \beta C^{\iota}$$

where k^0 is the initial RBF kernel that allows for the exploitation of dependencies on consecutive time points and β is how strongly we want networks that belong to the same cluster to be similar.
Given κ^{ι} we minimise the related $\text{TGL}_{\kappa^{\iota}}$, $\text{TIGM}_{\kappa^{\iota}}$ or $\text{PIGM}_{\kappa^{\iota}}$. We keep alternating the two minimisation steps until C^{ι} is equal to $C^{\iota-1}$.
The pseudo-code of this procedure is presented in Algorithm 3. Note that we could not use directly the similarity matrix S^{ι} to define the kernel to not rely too deeply on the first exploratory step.
We want to remark that, while we propose such automatic inference for temporal model, this approach can also be suitable for inference of both clusters and networks also in the case of multi-class problems (see Section 4.6).

4.3 Kernel Temporal Graphical Lasso

The Gaussian assumption is the most explored in literature given its theoretical properties that simplify the inference process. Therefore, in the state of the art we can find methods on GGMs that use temporal consistency, kernels or clustering. In particular temporal consistency is exploited to model multivariate time series in (Bianco-Martinez et al., 2016; Hallac et al., 2017a; Harutyunyan et al., 2019; Tomasi et al., 2018a,b). Hallac et al., 2017a also consider asynchronous observations, *i.e.*, where networks may be not all equally spaced in time. Kernels are also widely used to model temporal variables, for example, Gaussian or Wishart Processes (WP) infer the covariance matrix in time. By definition, this approach does not estimate sparse graphs, which is critical in high-dimensional settings when the number of variables exceeds the number of available samples ($N \ll D$) (Fox and West, 2011; Rasmussen, 2003;

Wilson and Ghahramani, 2011). Kernels were also used in (Chang, Yao and Allen, 2019) for the inference of latent temporal graphical models by computing a kernel-dependent covariance matrix.
In case of Gaussian data we consider a dynamic graphical model as a non-stationary zero-mean normal multivariate probability distribution

$$p_G = \left(\mathcal{N}_1(\mathbf{0}, K_1^{-1}), \ldots, \mathcal{N}_T(\mathbf{0}, K_T^{-1})\right)$$

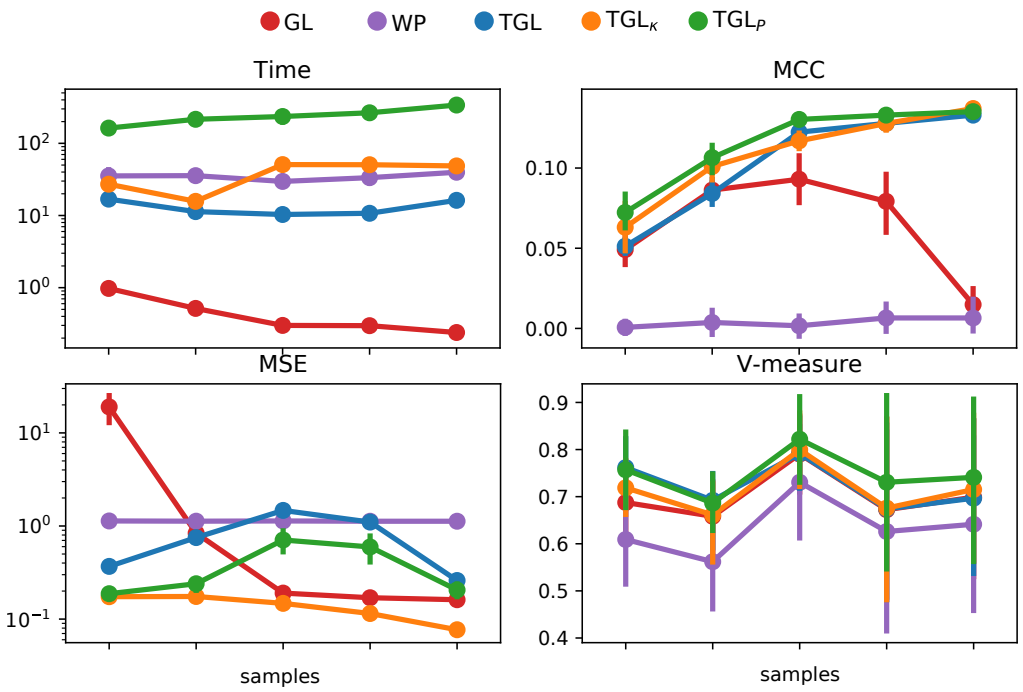
on $X_1, \ldots, X_D$ that factorises according to a graph G. The related TGL_κ model is then obtained from the model in Equation (30) instantiated with the Gaussian likelihood ℓ_{GGM} in Equation (11), as follows

$$\underset{\boldsymbol{K}\succ 0, \boldsymbol{K}\in\mathcal{S}^T}{\text{minimize}} \sum_{t=1}^{T} \left[-\ \ell_{GGM}(X_t|K_t) + \alpha \left\|K_t\right\|_{\text{od},1} \right] + P_{\Psi,\kappa}(\boldsymbol{K}) \tag{34}$$

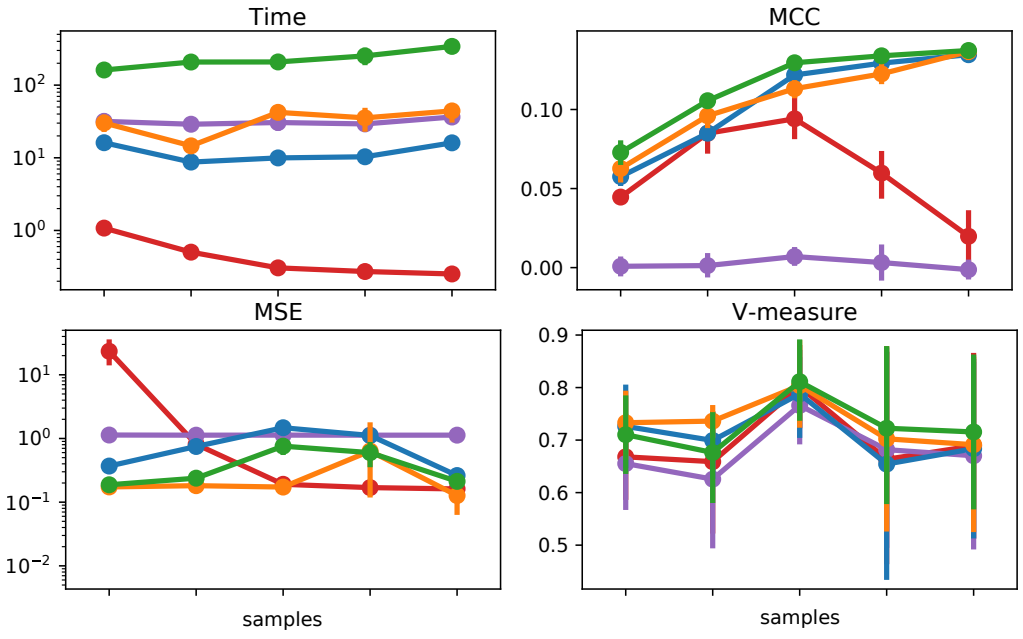
Note that in this case we also need a positive definite constraint, $\boldsymbol{K} \succ 0$ to ensure the log det function to be well-defined.

4.3.1 *Synthetic data experiments*

Here, we compare the performance of methods with the Markovian assumption and without it, *i.e.*, exploiting kernels. To this aim, we devised two synthetic experiments. In the first experiment, data show periodic temporal dependencies, while, in the other, data are characterised by random temporal dependencies. The data set used in the experiments followed the cluster-based generation schema (see Appendix C.4), with periodic and random pattern for $D = 100$ dimensions and $T = 20$ time points. For each data set we generated an increasing number of samples ($N_t \in \{5, 10, 50, 100, 500\}$) by sampling from the related multi-variate normal distribution. We generated the datasets 10 times for each experiment to perform stability evaluation.
In both cases, we evaluated the modelling performance of TGL_{ESS} (*i.e.*, TGL_κ with ESS periodic kernel), and TGL_P compared to the Time-Varying Graphical Lasso (TGL), Wishart Processes (WP) and the baseline graphical lasso (GL) (Friedman, Hastie and Tibshirani, 2008; Hallac et al., 2017a; Tomasi et al., 2018b; Wilson and Ghahramani, 2011). As our experiments are set in the $N \ll D$ scenario we could not consider TICC (Hallac et al., 2017b), the method that automatically clusters the networks.
The hyper-parameters of the methods (with the exception of WP) were selected through a stratified k-fold based on log-likelihood score (Molinaro, Simon and Pfeiffer, 2005) that relies on a Gaussian process-based Bayesian optimisation procedure (see Section 3.3.1). For all methods, we fixed the maximal amount of iterations at 500. For WP, which do not assume sparsity, we apply a threshold ($\epsilon = 10^{-5}$) to discard links with low weight and we take the maximum-a-posteriori estimate selected across the last 75% iterations (the first 25% were discarded for burn-in).
We computed the divergence from the ground truth in terms of structure of the network and edges weight. We also computed the V-measure to compare the



(A) *Periodic dependencies*



(B) *Random dependencies*

FIGURE 19. *Performance of Time (in seconds), Matthew Correlation Coefficient (MCC), Mean Squared Error (MSE) and V-measure for the Graphical Lasso (GL), Wishart Processes (WP), Time-varying Graphical Lasso (TGL), Kernel Temporal Graphical Lasso (TGL$_K$) and Temporal Graphical Lasso with Pattern detection (TGL$_S$) for two experiments on complex temporal dependencies: periodic dependencies (top results) and random dependencies (bottom results) on networks of $D = 100$ dimensions, $T = 20$ times and increasing sample size $N_t \in \{5, 10, 50, 100, 500\}$.*

inferred clusters with the ground truth (see Section 3.2). For TGL_{ESS}, TGL, WP and GL, which do not include a way to cluster graphs over time, we estimated a V-measure by first inferring the dynamic network and then run a clustering algorithm a posteriori. We provide an indication on the average time of convergence for each method, to demonstrate the scalability of TGL_{ESS} and TGL_{P}.

4.3.2 *Results*

TGL_{ESS} and TGL_{P} perform better than the methods that do not consider non-Markovian temporal dependencies. TGL_{P} recovers the clusters with high accuracy. Figure 19 shows a visual representation of the different inference methods in the two experimental settings, where we averaged the results across repetitions. Results are visualised for an increasing number of samples available at each time point. In Table 1 we zoom in and show more details on the performance for $N_t = 50$ available samples. Both cases show how a temporal kernel is beneficial to better infer the underlying system structure. Indeed, TGL_{ESS} and TGL_{P} outperform the competitors in terms of MCC, MSE and V-measure, especially when $N \ll D$.
TGL_{ESS} has high performance scores when data exhibit a periodic pattern (Figure 19a). Indeed, prior information on the behaviour of the network is crucial for a reliable inference, fact that is reflected in the improvement of precision and MSE. Also, TGM_{ESS} converges in the same amount of time as TGL even though it considers a higher number of relations between networks. TGM_{P} requires more time to converge due to the two-step alternating minimisation procedure. Nonetheless, TGL_{P} outperforms in almost all measures the other methods, including TGL_{ESS}, by increasing the accuracy in structure inference and reducing the error in the estimation of the dynamical network. Both Figure 19 and table 1 show for TGL_{P} high V-measure which indicates how automatic pattern discovery is appropriate to detect both random and non-random patterns.

4.4 Temporal Ising Graphical Models

The Ising Graphical model (IGM) has been studied in the stationary case in (Ravikumar, Wainwright and Lafferty, 2010; Yang et al., 2015). Based on the stationary models, methods that studied its temporal evolution have been proposed in the context of neural spikes modelling. In particular Roudi, Tyrcha and Hertz, 2009 proposed a method that, for neural spikes time series, retrieves a unique stationary connectivity model. Later on, Hertz, Roudi and Tyrcha, 2011 proposed a temporal Ising model based on the combination of standard graphical model coupled with differential equations that model the dynamic of neurons. The proposed algorithms, that infer respectively a stationary and a non-stationary network, have the limitation of focusing on neural spike activity and thus embedding a specific dynamic in the model. As we

experiment	method	BA	P	MCC	MSE	V-measure
(a) Periodic-pattern	GL	0.505 ± 0.002	0.029 ± 0.003	0.093 ± 0.017	0.190 ± 0.003	0.790 ± 0.096
	TGL	0.558 ± 0.005	0.301 ± 0.014	0.122 ± 0.003	1.470 ± 0.015	0.791 ± 0.089
	WP	0.498 ± 0.002	0.022 ± 0.001	0.002 ± 0.007	1.133 ± 0.003	0.730 ± 0.137
	TGL_κ	0.560 ± 0.008	$\mathbf{0.372 \pm 0.046}$	0.117 ± 0.005	$\mathbf{0.148 \pm 0.007}$	0.800 ± 0.090
	TGL_P	$\mathbf{0.577 \pm 0.003}$	0.341 ± 0.014	$\mathbf{0.130 \pm 0.002}$	0.707 ± 0.211	$\mathbf{0.822 \pm 0.113}$
(b) Random-pattern	GL	0.505 ± 0.001	0.029 ± 0.003	0.094 ± 0.014	0.191 ± 0.002	0.802 ± 0.088
	TGL	0.560 ± 0.004	0.299 ± 0.011	0.122 ± 0.004	1.475 ± 0.012	0.788 ± 0.093
	WP	0.497 ± 0.003	0.022 ± 0.001	0.007 ± 0.005	1.130 ± 0.006	0.766 ± 0.099
	TGL_κ	0.553 ± 0.007	0.284 ± 0.033	0.113 ± 0.005	$\mathbf{0.173 \pm 0.017}$	0.804 ± 0.088
	TGL_P	$\mathbf{0.574 \pm 0.004}$	$\mathbf{0.331 \pm 0.015}$	$\mathbf{0.130 \pm 0.003}$	0.758 ± 0.181	$\mathbf{0.811 \pm 0.081}$

TABLE 1. *Performance in terms of balanced accuracy (BA) average precision (P), Matthews correlation coefficient (MCC), mean squared error (MSE) and V-measure for TGL_κ and TGL_S with respect to GL (baseline), TGL and WP for a graph of $D = 100$ nodes with $N_t = 50$ samples per $T = 20$ time points.*

introduced in previous chapters, this thesis work is not application-driven but aims at tackling the temporal aspect of processes. Therefore, we extend the Ising model to include a temporal evolution without using prior knowledge on a specific domain. Indeed, even if the Ising model is the most appropriate model for neural signals (Schneidman et al., 2006) it can also be suitable for the modelling of other types of data as voting patterns (Banerjee, Ghaoui and d'Aspremont, 2008), single nucleotide genetic mutations, behaviour of gases or magnets (Ising, 1925) and many others. Moreover, the simultaneous clustering and inference of networks would allows us to detect repetitions patterns without imposing any prior knowledge.

We consider a dynamic graphical model as a non-stationary probability distribution

$$p_G(X_1, \dots, X_D | \boldsymbol{K}) = \big(p_{IGM}(K_1), \dots, p_{IGM}(K_T)\big)$$

on $X_1, \dots, X_D$ that factorises according to the graph G. The TIGM_κ model takes the form in (30) instantiated with the Ising conditional likelihood ℓ_{IGM} in Equation (15) and it is defined as

$$\underset{\boldsymbol{K} \in \mathcal{S}^T}{\text{minimize}} \sum_{t=1}^{T} \Big[- \ell_{IGM}(X_t | K_t) + \alpha \left\| K_t \right\|_{\text{od},1} \Big] + P_{\Psi,\kappa}(\boldsymbol{K}) \tag{35}$$

Note that, in this model the imposition of a symmetry constraint is fundamental as we cannot simply reason in terms of single variables because of the penalty $P_{\Psi,\kappa}$.
An example of a possible application of TIGM_κ is presented in the context of neural data in Figure 23. Here, we generated a simulation of neural activity following a repeating networks pattern. By observing the spikes on different neurons it is really difficult to observe a consistent temporal behaviour between

time spans that have the same underlying network (blue cluster). Thus, we could employ inference techniques to detect similarities of networks in time.

4.4.1 *Synthetic data experiments*

The evaluation of TIGM$_\kappa$ is performed via three experiments. For all experiments we sampled the N_t independent observations from the generated dynamic network using the Metropolis-Hastings algorithm where each sample is generated after 100 repetitions for burn-in (Epskamp, 2015). We selected the hyper-parameters using the stability-based model selection strategy proposed in Chapter 3.

STATIONARY MINIMISATION COMPARISON

We consider the stationary case and we want to assess which minimisation algorithm to use for the optimisation of the Ising model considering the following three strategies:

(a) Single-FBS: a Forward Backward Splitting (FBS) procedure that performs neighbourhood selection separately on the D variables, and then unifies the neighbourhoods in a post-processing step (Allen and Liu, 2013; Ravikumar, Wainwright and Lafferty, 2010).

(b) Logistic Regression: a Logistic regression problem that similarly is applied on the D variables considering in turn one variable as the output and the remaining $D-1$ as independent covariates; again the final network is retrieved in a post-processing step (Ravikumar, Wainwright and Lafferty, 2010; Wan et al., 2016).

(c) Global-FBS: a FBS procedure that simultaneously optimises on the entire adjacency matrix through the imposition of symmetry constraint on the solution.

We generated a stationary network using the Networkx package (Hagberg, Swart and S Chult, 2008) with $D = 20$ nodes and $N = 100$ samples and we repeated the experiment 10 times to obtain mean values of three minimisation algorithms.

STATIC VS TEMPORAL COMPARISON

The second experiment aims at assessing the goodness of the temporal model with respect to the static one.

We performed three experiments with an increasing number of variables $D = \{5, 10, 50\}$ keeping N_t fixed to 100 for $T = 10$ times for a totality of $T(D(D-1)/2)$ unknowns using an ℓ_1 generation schema (see Appendix C.1). We generated data sets 10 times and fit the static IGM (with Globlal-FBS optimisation) and TIGM$_\kappa$ instantiated with an RBF kernel.

method	Global-FBS	Logistic Regression	Sigle-FBS
P	0.73 ± 0.04	0.71 ± 0.01	$\mathbf{0.76 \pm 0.06}$
R	$\mathbf{0.76 \pm 0.1}$	0.76 ± 0.06	0.57 ± 0.1
F_1	$\mathbf{0.74 \pm 0.04}$	0.73 ± 0.03	0.65 ± 0.06
S	0.4 ± 0.15	0.37 ± 0.13	$\mathbf{0.62} \pm 0.18$
BA	0.58 ± 0.04	0.56 ± 0.04	$\mathbf{0.6} \pm 0.05$
MCC	$\mathbf{0.48} \pm 0.11$	0.46 ± 0.11	0.45 ± 0.1
Time(s)	10.34 ± 0.27	$\mathbf{0.04 \pm 0.01}$	91.98 ± 3.4

TABLE 2. *Performance in terms of Precision (P), Recall (R), F_1 score (F_1), Specificity (S), Balanced Accuracy (BA), Matthews Correlation Coefficient (MCC) and Time in second for different minimisation algorithms for the stationary Ising model, in particular the Global-FBS, Logistic Regression and Single-FBS on a network of $D = 20$ nodes with $N = 100$ samples.*

PRIOR KERNELS VS PATTERN DETECTION

The third experiment assesses the improvement we obtain using a kernel and, thus, taking advantage of a strong prior as well as the ability of automatic pattern dependencies discovery.
We generated cluster-based networks (see Appendix C.4) with 3 cluster representatives of $D = 10$ variables in a $T = 15$ dynamic network with a periodic pattern. We sampled $N_t = 100$ observations for each time point.

4.4.2 *Results*

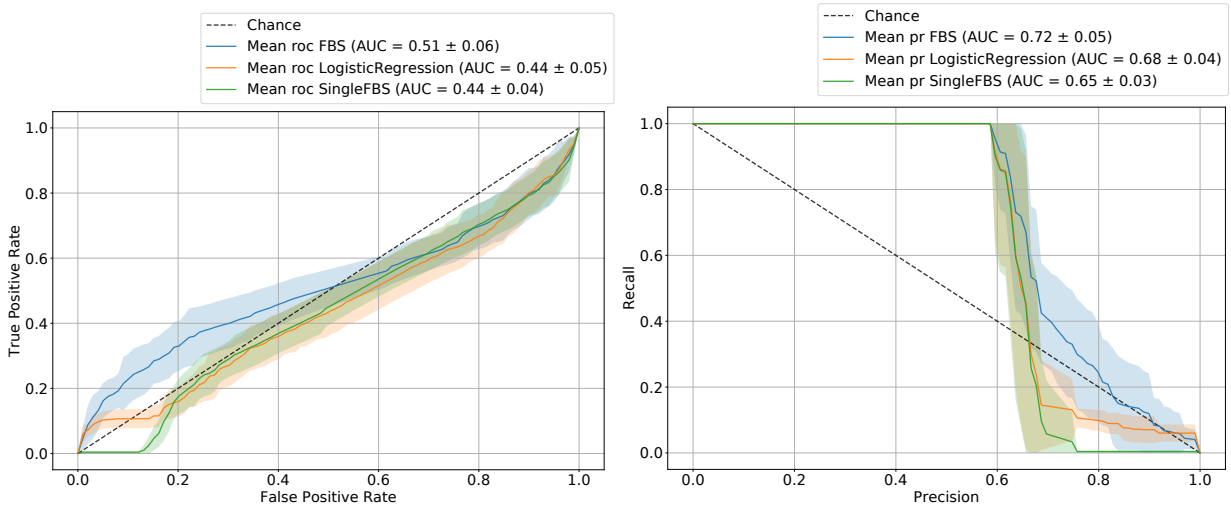
We assessed the results only in terms of structure recovery as considering the problem as a regression task (see Section 3.2) is not meaningful. Indeed, the Ising model only has edges with values $\{-1, 0, 1\}$.

STATIONARY MINIMISATION COMPARISON

Global-FBS performs better than the other minimisation methods despite requiring more computational time than Logistic Regression. In Table 2 and Figure 20 we show the results obtained comparing optimisation methods for stationary inference. Global-FBS optimisation procedure is the one that overall has the best performance. Indeed, from the execution time point of view Global-FBS is not faster than the Logistic Regression but it produces more accurate results in terms of recovery of the network structure. Single-FBS has higher precision and specificity with a consequently higher balanced accuracy but the results for Global-FBS are comparable. The ROC and PR curves seems to indicate the same conclusion as, for both, Global-FBS has a higher mean AUC (0.51 and 0.72 respectively).

STATIC VS TEMPORAL COMPARISON

$TIGM_\kappa$ performs better than its stationary counterpart especially when the number



(A) *ROC curve* (B) *Precision-Recall curves*

FIGURE 20. *Average results across 10 repetitions in terms of ROC and PR curves for the application of different minimisation algorithm on stationary Ising model with* $N = 100$, $D = 120$ *for a fixed hyper-parameter* α.

of variables increases with respect to the available samples. We compared TIGM_{RBF} fixing $\psi = \ell_1$ with IGM. Results are presented in Table 3 and Figure 24. We observe that the ROC and PR curves show a consistently higher performance on the temporal implementation with respect to the stationary case, which is also supported by a smaller time to convergence of different order of magnitudes (see last row of Table 3). If we observe the structure performance in Table 3 we note that results are consistent as the number of variables increases. Also, even if recall and F_1 score are higher for the stationary Ising model, we still have comparable results especially when the number of variables increases. TIGM_{κ} has significantly higher scores when looking at specificity.

PRIOR KERNELS VS PATTERN DETECTION

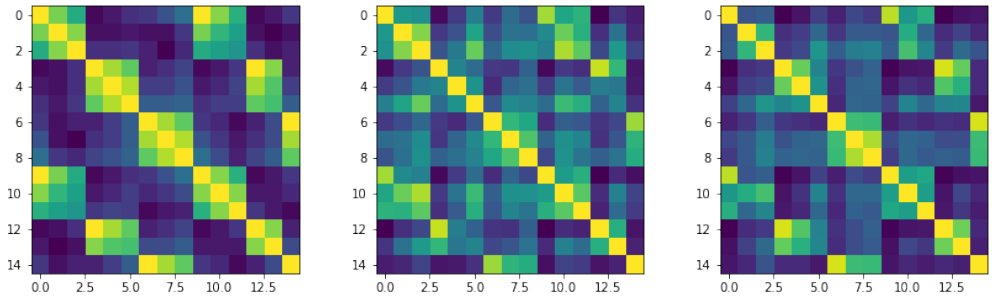
TIGM_{ESS} better recovers the structure while TIGM_{P} better recovers the dependency pattern. We compared TIGM_{ESS} with TIGM_{P} for the recovery of both structure and clusters with a periodical recurrence. Structure recovery performances (Table 4) are highly similar but we can still observe a slightly higher performance for TIGM_{ESS} given by the true prior imposed on the solution. In turn, by looking at Figure 21 we can observe that TIGM_{P} produces a better estimate of the true clusters of networks (*i.e.*, networks with similar and thus dependent structure).

4.5 Temporal Poisson Graphical Models

The Poisson Graphical Model (PGM) has been studied in different state-of-the-art papers in the stationary case (Allen and Liu, 2013; Yang et al., 2012, 2013, 2015). Nonetheless, to the best of our knowledge it was never extended

	5 variables		10 variables		50 variables	
Metric	IGM	TIGM_{RBF}	IGM	TIGM_{RBF}	IGM	TIGM_{RBF}
P	0.71 ± 0.06	$\mathbf{0.78 \pm 0.05}$	0.69 ± 0.08	$\mathbf{0.77 \pm 0.09}$	0.73 ± 0.02	$\mathbf{0.74 \pm 0.02}$
R	$\mathbf{0.89 \pm 0.05}$	0.59 ± 0.09	$\mathbf{0.90 \pm 0.04}$	0.70 ± 0.16	$\mathbf{0.49 \pm 0.03}$	0.46 ± 0.03
F_1	$\mathbf{0.78 \pm 0.02}$	0.66 ± 0.05	$\mathbf{0.78 \pm 0.04}$	0.71 ± 0.06	$\mathbf{0.59 \pm 0.03}$	0.57 ± 0.03
S	0.18 ± 0.05	$\mathbf{0.61 \pm 0.16}$	0.17 ± 0.09	$\mathbf{0.50 \pm 0.28}$	0.64 ± 0.02	$\mathbf{0.69 \pm 0.03}$
BA	0.53 ± 0.02	$\mathbf{0.60 \pm 0.06}$	0.53 ± 0.03	$\mathbf{0.60 \pm 0.06}$	0.56 ± 0.02	$\mathbf{0.57 \pm 0.02}$
Time(s)	$1.5e10 \pm 5.3e2$	$\mathbf{5.7e2 \pm 4.8e1}$	$1.5e10 \pm 4.9e2$	$\mathbf{5.2e2 \pm 1.2e1}$	$1.5e10 \pm 4.7e2$	$\mathbf{5.8e2 \pm 9.3e1}$

TABLE 3. *Performance in terms of Precision (P), Recall (R), F_1 score (F_1), Specificity (S), Balanced Accuracy (BA), Matthews Correlation Coefficient (MCC) and Time in second for Kernel Temporal Ising Graphical Model (TIGM_{RBF}) and stationary Ising Graphical Model (IGM) for an increasing number of variables $D = \{5, 10, 50\}$ with a fixed number of samples $N_t = 100$.*



(A) *True dependency pattern* (B) *Inferred pattern with* TIGM_{ESS} (C) *Inferred pattern with* TIGM_P

FIGURE 21. *Qualitative comparison of automatic dependency patterns inference for Temporal Ising Graphical Model with periodic kernel (TIGM_{ESS}) (panel b) and Temporal Ising Graphical Model with Pattern detection (TIGM_P) (panel c) compared to the ground truth (panel a) for the inference of dependency pattern with periodical repetitions in $T = 15$ time points on $D = 10$ variables and $N_t = 100$ observations.*

to consider a longitudinal component. We argue that it would be extremely interesting to have such temporal model given its natural predisposition to handle counts data, such as in the case of NGS sequencing data (Metzker, 2010). While current literature solves the problem by using GGMs after suitable transformation, a temporal Poisson graphical models would be better suited to model dynamical biological data that are intrinsically not Gaussian.
We consider a dynamic graphical model as a non-stationary probability distribution

$$p_G(X_1, \ldots, X_D | \boldsymbol{K}) = \left(p_{PGM}(K_1), \ldots, p_{PGM}(K_T)\right)$$

on $X_1, \ldots, X_D$ that factorises according to the graph G. The TPGM_κ model is then defined as Equation (30) instantiated with the Poisson conditional likelihood ℓ_{PGM} in Equation (17).

Score	TIGM_{ESS}	TIGM_{P}
P	**0.71**	0.71
R	**0.61**	0.49
F_1	**0.66**	0.58
S	0.57	**0.65**
BA	**0.59**	0.57

TABLE 4. *Performance in terms of Precision (P), Recall (R), F_1-score (F_1), Specificity (S), Balanced Accuracy (BA) and time in seconds of Kernel Temporal Ising Graphical Model (TIGM_{ESS}) and the Temporal Ising Graphical Model with Pattern detection (TIGM_{P}) for the inference of networks with periodical repetitions in $T = 15$ time points on $D = 10$ variables and $N_t = 100$ observations.*

	5 nodes		10 nodes		20 nodes	
Metric	**Static**	**Temporal**	**Static**	**Temporal**	**Static**	**Temporal**
P	$\mathbf{0.97 \pm 0.05}$	0.96 ± 0.05	$\mathbf{0.75 \pm 0.08}$	0.73 ± 0.07	$\mathbf{0.56 \pm 0.02}$	0.50 ± 0.02
R	0.88 ± 0.1	$\mathbf{0.89 \pm 0.09}$	0.66 ± 0.09	$\mathbf{0.77 \pm 0.08}$	0.66 ± 0.04	$\mathbf{0.80 \pm 0.04}$
F_1	0.92 ± 0.07	$\mathbf{0.92 \pm 0.06}$	0.70 ± 0.06	$\mathbf{0.74 \pm 0.04}$	0.60 ± 0.02	$\mathbf{0.61 \pm 0.02}$
S	0.96 ± 0.02	$\mathbf{0.98 \pm 0.02}$	$\mathbf{0.85 \pm 0.07}$	0.80 ± 0.09	$\mathbf{0.81 \pm 0.01}$	0.72 ± 0.03
BA	0.93 ± 0.06	$\mathbf{0.94 \pm 0.05}$	0.75 ± 0.04	$\mathbf{0.79 \pm 0.04}$	0.74 ± 0.02	$\mathbf{0.76 \pm 0.01}$
Time	$1.5e9 \pm 2.3e3$	$\mathbf{2.7e3 \pm 1.4e2}$	$1.5e9 \pm 2.2e3$	$\mathbf{2.7e3 \pm 8.8e1}$	$1.5e10 \pm 2.3e24$	$\mathbf{2.7e3 \pm 6.1e1}$

TABLE 5. *Performance in terms of Precision (P), Recall (R), F_1-score (F_1), Specificity (S), Balanced Accuracy (BA) and time in seconds of Temporal Poisson Graphical Model (TPGM_{RBF}) against stationary Poisson Graphical Model (PGM) for an increasing number of variables $D = \{5, 10, 50\}$ with a fixed number of samples $N_t = 100$ and $T = 10$ time points.*

$$\underset{\boldsymbol{K} \in \mathcal{S}^T}{\text{minimize}} \sum_{t=1}^{T} \left[- \ell_{PGM}(X_t | K_t) + \alpha \left\| K_t \right\|_{\text{od},1} \right] + P_{\Psi,\mathcal{K}}(\boldsymbol{K}) \tag{36}$$

This model requires the symmetry constraint given the imposition of the penalty $P_{\Psi,\mathcal{K}}$.

4.5.1 *Synthetic data experiments*

We performed one experiment with an increasing number of variables $D = \{5, 10, 20\}$ keeping $N_t = 100$ for $T = 10$. For each D we used and ℓ_1 evolution schema letting three edges change per each time (see Appendix C.1). For the generation of the N_t independent observations we used the approach proposed by (Allen and Liu, 2013; Karlis, 2003). At each time t, the sample matrix $X_t \in \mathcal{X} = \mathbb{N}^{N_t \times D}$ is generated by the following model

$$X_t = YB_t + E$$

where $Y \in \mathbb{N}^{N \times D(D-1)}$ such that each element of the matrix $Y_{ij} \sim \text{Poisson}(\lambda^*)$ and $E \in \mathbb{N}^{N_t \times D}$ is such that $E_{ij} \sim \text{Poisson}(\lambda_{noise})$. The matrix B encodes the true underlying graph structure denoted by the adjacency matrix $K \in \{0,1\}^{D \times D}$ such that

$$B_t = \left[I_D; P \odot (1_D \text{tri}(K_t)^\top) \right]^\top$$

where P is the $D \times (D(D-1)/2)$ pair-wise permutation matrix, $\odot$ denotes the Hadamard or element-wise product and $\text{tri}(K_t)$ denotes the $D(D-1)/d \times 1$ vectorized upper triangular portion of the adjacency matrix K_t. We fixed a high signal-to-noise level $\lambda^* = 1$, $\lambda_{noise} = 0.5$. We compare PGM with TPGM_{RBF} with $\Psi = \ell_1$ after selecting the hyper-parameters using the stability-based model selection strategy proposed in Chapter 3.

4.5.2 *Results*

TPGM_{RBF} performs better than PGM as D increases compared to the number of samples N_t. Results are presented in Table 5 and Figure 25. By looking at the table we observe that TPGM_{RBF} always requires less time to converge than stationary PGM applied on the 10 networks. We can also observe that with 5 nodes TPGM_{RBF} has a better performance in structure recovery according to all the measure while, for 10 and 20 nodes, the stationary PGM has a slightly higher precision and specificity. Nonetheless, if we look at the ROC and Precision-Recall curves in Figure 25 we can observe that the behaviour of the two methods with 5 nodes is the same in terms of balances between True Positive and False Positive rates as well as in terms of precision-recall balance. As the number of nodes increases the temporal models performs better than the stationary one (higher AUC scores).
In the Precision-Recall curves (Figure 25) it is possible to observe a fast drop of Recall that remains then stable. This is due to the diagonal presence in the matrix, indeed the diagonal values are typically higher than the non-diagonal elements. Therefore, when the thresholds are computed there is a certain point in which only the diagonal element remains, this causes a drop in recall while the precision remains stable. This behaviour is shown in Figure 22, where we can observe the drop in recall after a certain threshold.

4.6 Multi-class problem

During this chapter, we heavily discussed the problem of inferring graphical models from temporal data where each time points has its own distribution. A strongly related problem is the one of inferring networks from samples belonging to different classes, that may have close but different underlying distributions. Such problem has been previously tackled in (Danaher, Wang and Witten, 2014; Guo et al., 2011) in the context of GGMs. They suggested that two graphical models inferred from different classes of the same population should be similar to one another as they share the most common structure

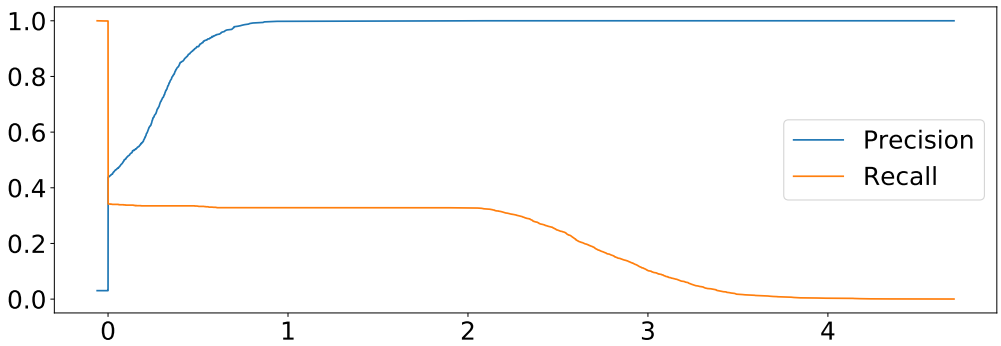


FIGURE 22. *Exemplification of the elbow behaviour in the Precision-Recall curves.*

with some differences that differentiate them into classes. Danaher, Wang and Witten, 2014 proposed the Joint Graphical Lasso (JGL), a method that jointly estimates multiple graphical models. They imposed a similarities between these models trough a penalty, as follows

$$\underset{\boldsymbol{K}\succ 0}{\text{minimize}} \sum_{t=1}^{T}\left[-\ell_{GGM}(X_t|K_t)+\alpha\left\|K_t\right\|_{\text{od},1}\right]+\beta\sum_{t=1}^{T}\sum_{m>t}\Psi(K_t-K_m).$$

Note, that this model corresponds exactly to our general model in Equation (30) in the case of Gaussian assumption and with the following kernel:

$$\kappa_{MC}(t_i,t_j)=\begin{cases}0, & \text{if } t_i=t_j\\ \beta, & \text{if } t_i\neq t_j\end{cases}$$

Therefore, our general model besides providing an increased modelling power in the context of temporal models also allow to extend the Ising and Poisson model easily to the multi-class case without introducing a further optimisation procedure.

4.6.1 *Synthetic experiments*

We provide preliminary results on the ability of TGL_{MC}, TIGM_{MC} and TPGM_{MC} to infer multi-class networks. We devised two experiments, Random-Graph where we generated the initial network on $D = 10$ nodes using the Erdős-Rényi algorithm (Erdős and Rényi, 1960) and the Preferential-Attachment where we generated the initial network following the Barabasi-Albert (Albert and Barabási, 2002) random model. The generation of $K = 5$ classes is performed following the schema presented in Appendix C.6. For each time point we generated an increasing number of sample $N_t \in \{5, 10, 50, 100\}$ to assess if structure recovery improves as the samples become more. We repeated the experiments 10 times to check for stability of the results. We applied on these data TGL_{MC}, TIGM_{MC} and TPGM_{MC} with $\Psi = \ell_1$ and $\Psi = \ell_1$, where we suitable selected the values of the edges and the sampling process according to the distribution. For all the models we fixed the β of the kernel to 1 and we fixed the hyper-parameters α.

4.6.2 *Results*

Results are presented in Figure 26 and Figure 27 for the Random-Graph and the Preferential-Attachment, respectively.

RANDOM-GRAPH
TIGM$_{MC}$ with ℓ_1 consistency performs better than the other methods and the other type of consistency. Figure 26 we observe that TIGM$_{MC}$ with ℓ_1 consistency has MCC that goes from 0.25 to 0.75 as the number of available samples increases. TGL$_{MC}$ and TPGM$_{MC}$ have a flat trend where TPGM$_{MC}$ performs slighlty better. The ℓ_2 consistency produces worst results for all methods.

PREFERENTIAL-ATTACHMENT
TPGM$_{MC}$ performs better than TGL$_{MC}$ and TIGM$_{MC}$ independently from the type of consistency applied. In Figure 27 we observe that all methods perform equally on the data independently on the type of consistency Ψ. TPGM$_{MC}$ model performs significantly better than the other two models which show an MCC score close to zero. Moreover, TGL$_{MC}$ and TIGM$_{MC}$ have flat trends of scores as the number of samples increases differently from TPGM$_{MC}$.

We want to remark that we kept the hyper-parameters fixed for all the models, therefore a suitable model selection procedure on the specific model could improve its performances. We plan on further assessing this extension in future work.

4.7 Summary

In this chapter we presented a generalised temporal model that allows to be flexible in terms of probability assumptions, temporal consistency and patterns of dependencies. We instantiated such models with three different likelihoods, Gaussian, Bernoulli and Poisson. We presented thorough validation on TGL$_\kappa$ and preliminary synthetic validation of TIGM$_\kappa$ and TPGM$_\kappa$. We also showed that our general model can be used for the inference of multi-class problems providing a single way to modelling and optimising on a variety of possible real-world data. We plan to further validate our models with a bigger comparison in terms of imposed kernels as well as a deeper assessment of the simultaneous inference of temporal similarities. We also need to check for more signal-to-noise ratio settings in the Poisson model. Furthermore, we plan to extend the current sequential implementation to consider parallel tasks that will further speed up the time needed for convergence.
We would like to remark that the general model proposed in Equation (30) could be instantiated, in principle, with all the possible likelihood allowing for the consideration of mixed graphical models (Yang et al., 2014) as well as for integrated one (Žitnik and Zupan, 2015). Indeed, the modular optimisation presented in Appendix A is easily extendible to any problem minimised

through coordinate descent, FBS or ADMM procedures. This would allow to easily consider both temporal and multi-class extensions of many network inference methods.

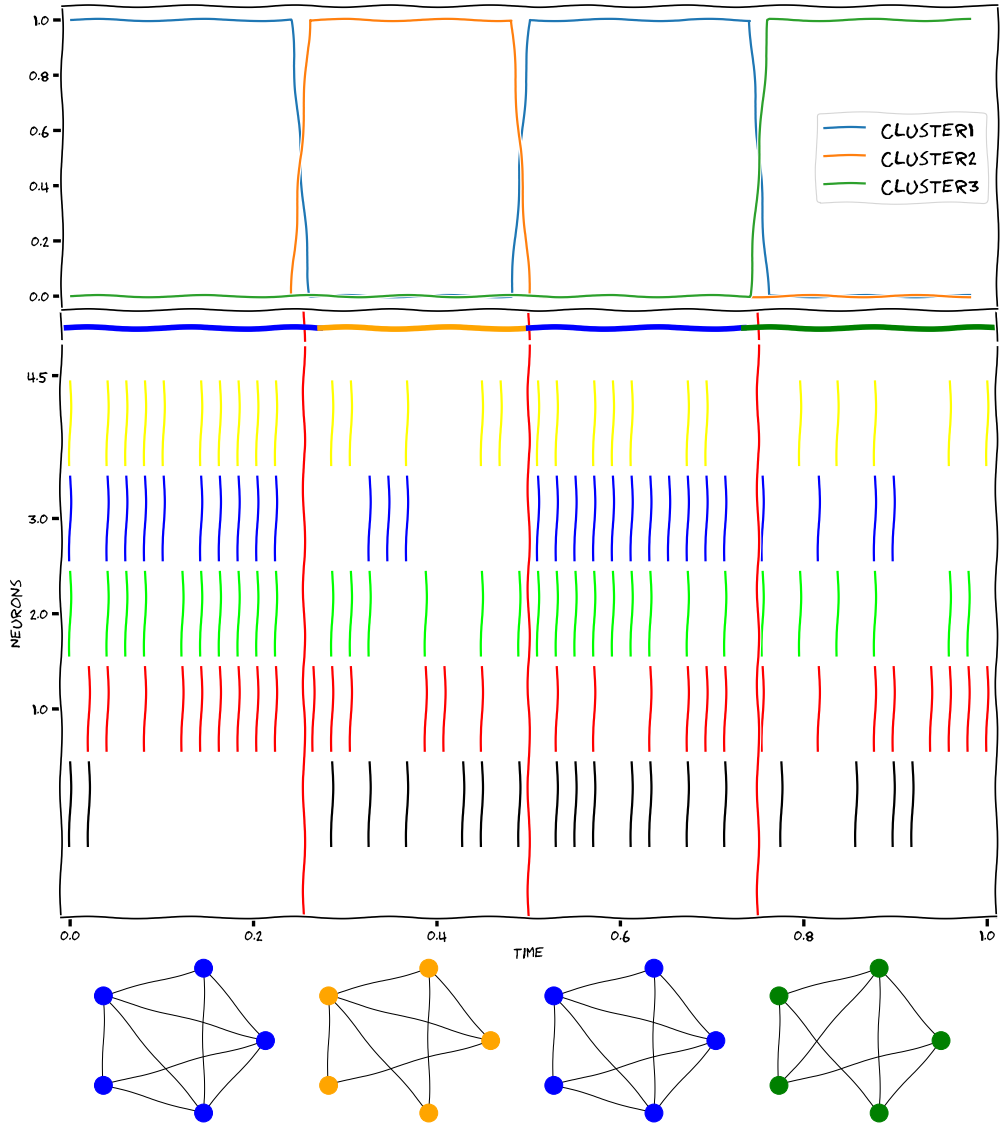
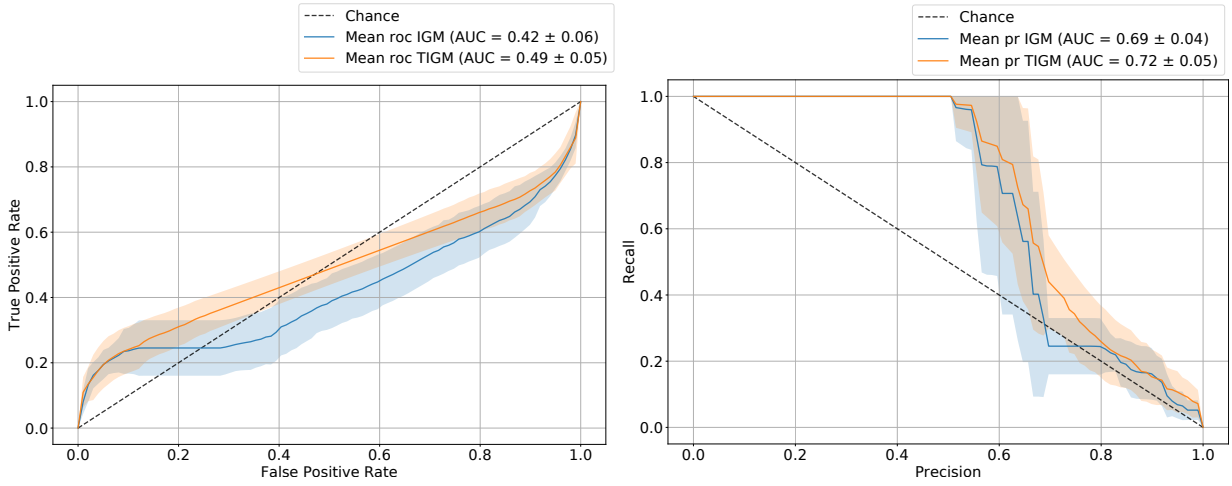
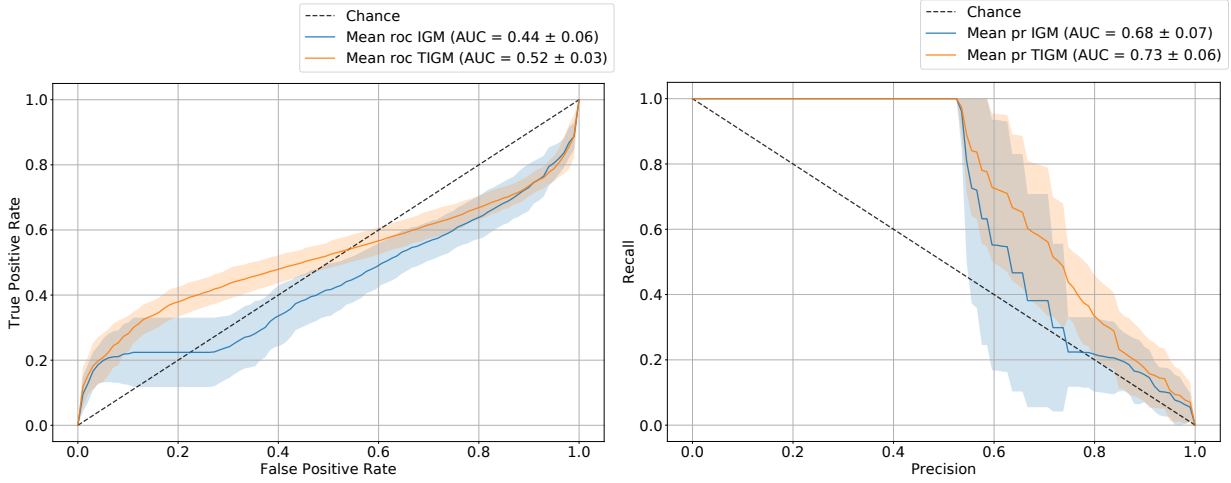


FIGURE 23. *Example of 5 neurons activity whose underlying connectivity network is clustered in time (clusters are shown at the top). By looking at the behaviour of clustered time points we cannot observe any significative resemblance but the inference of the underlying networks (plotted on the bottom) guides us in the detection of similarities.*



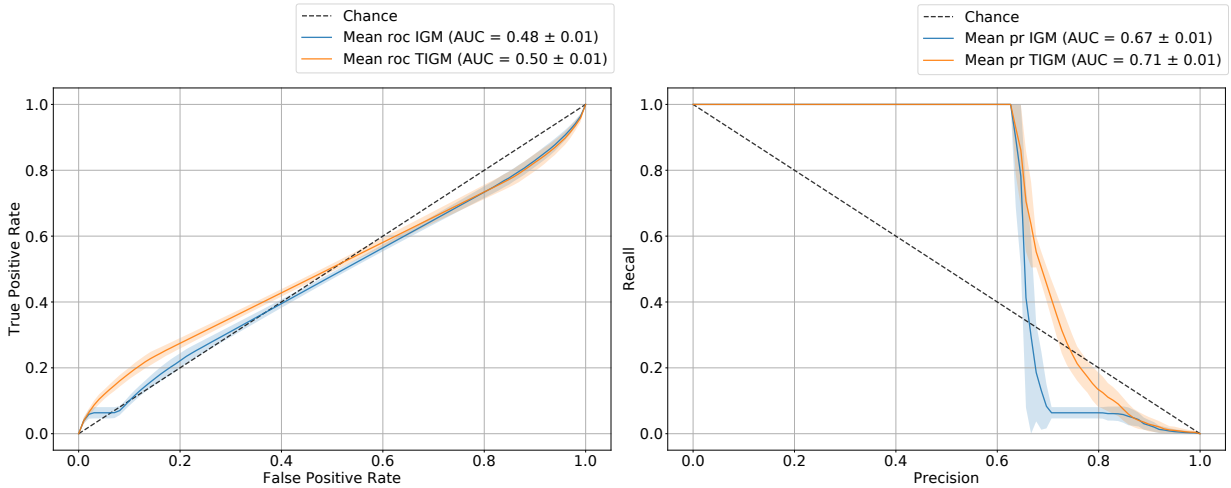
(A) *ROC curve 5 nodes*

(B) *Precision-Recall curve 5 nodes*



(C) *ROC curve 10 nodes*

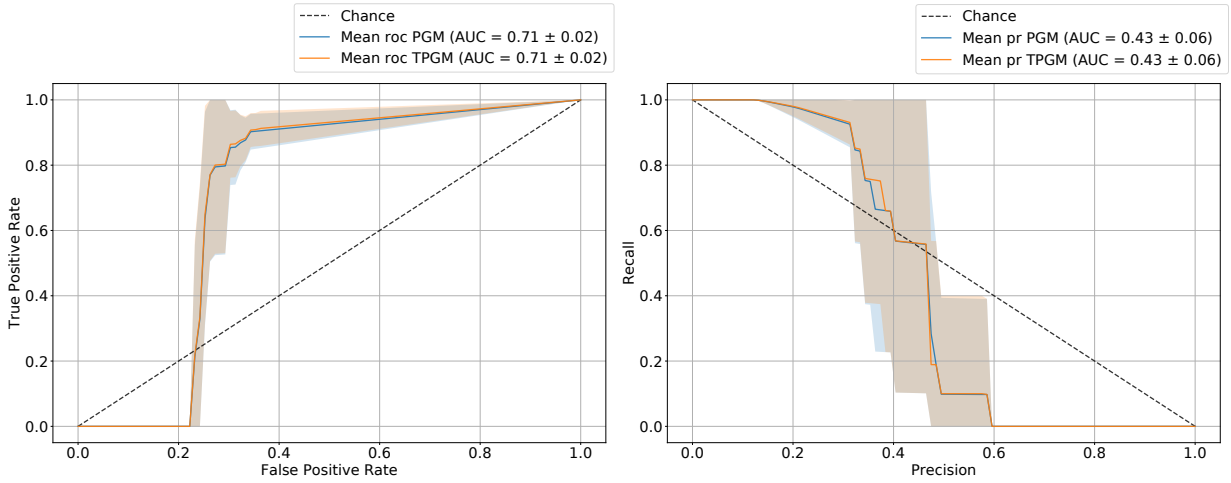
(D) *Precision-Recall curve 10 nodes*



(E) *ROC curve 50 nodes*

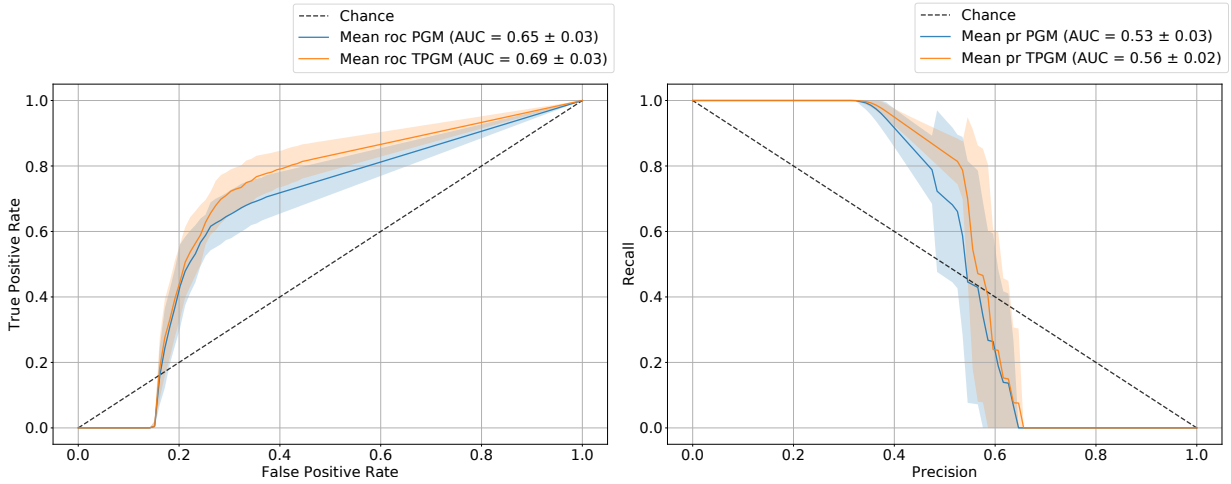
(F) *Precision-Recall curve 50 nodes*

FIGURE 24. *Average results across 10 repetitions in terms of ROC and PR curves for the comparison of Temporal Ising Graphical Model with RBF kernel ($TIGM_{RBF}$) against stationary Ising Graphical Model (IGM) for an increasing number of variables $D = \{5, 10, 50\}$ at $T = 10$ time points with a fixed number of samples $N_t = 100$.*



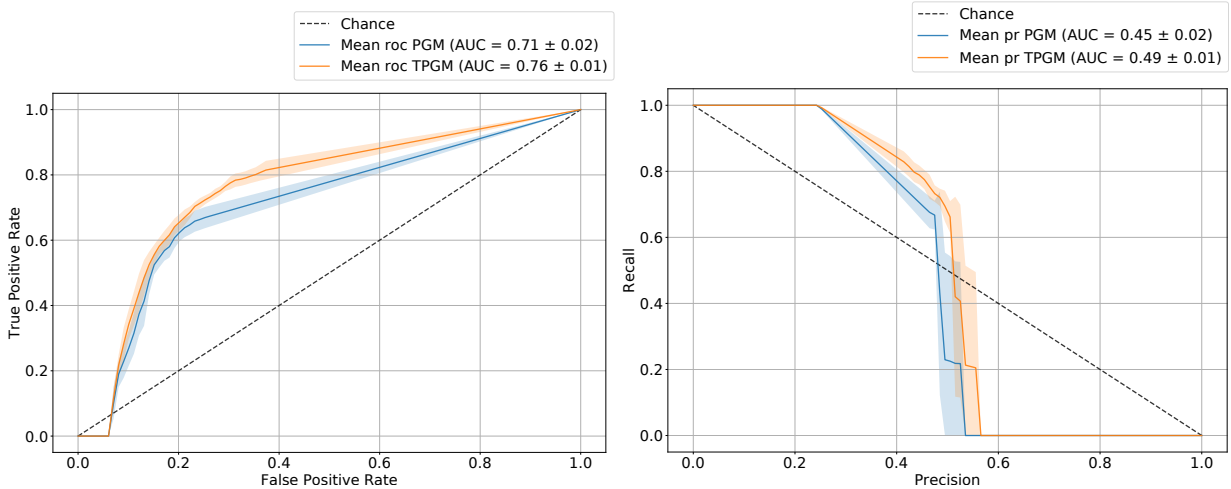
(A) *ROC curve 5 nodes*

(B) *Precision-Recall curve 5 nodes*



(C) *ROC curve 10 nodes*

(D) *Precision-Recall curve 10 nodes*



(E) *ROC curve 20 nodes*

(F) *Precision-Recall curve 20 nodes*

FIGURE 25. *Average results across 10 repetitions in terms of ROC and PR curves for the comparison of Temporal Poisson Graphical Model with RBF kernel ($TPGM_{RBF}$) against stationary Poisson Graphical Model (PGM) for an increasing number of variables $D = \{5, 10, 20\}$ at $T = 10$ time points with a fixed number of samples $N_t = 100$.*

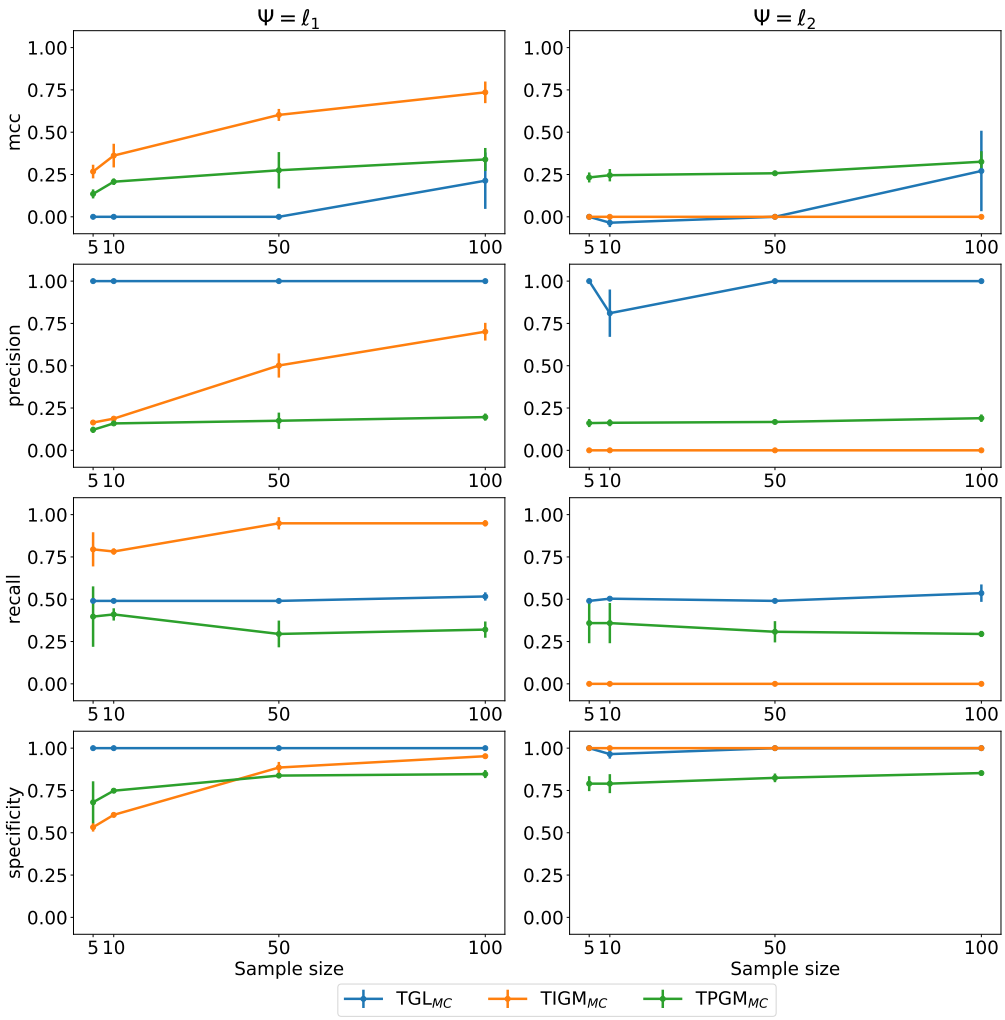


FIGURE 26. *Average results across 10 repetitions in terms of Matthew Correlation Coefficient (mcc, first row), precision (second row), recall (third row) and specificity (bottom row) for the comparison of Temporal Graphical Lasso with Multi-Class kernel (TGL_{MC}), Temporal Ising Graphical Model with Multi-Class kernel ($TIGM_{MC}$) and Temporal Poisson Graphical Model with Multi-Class kernel ($TPGM_{MC}$) at increasing sample size $N_t \in \{5, 10, 50, 100\}$ with $D = 10$ variables and 5 classes for multi-class experiments with Erdő-Rényi random networks in case of ℓ_1 (first column) and ℓ_2 (second column) consistencies.*

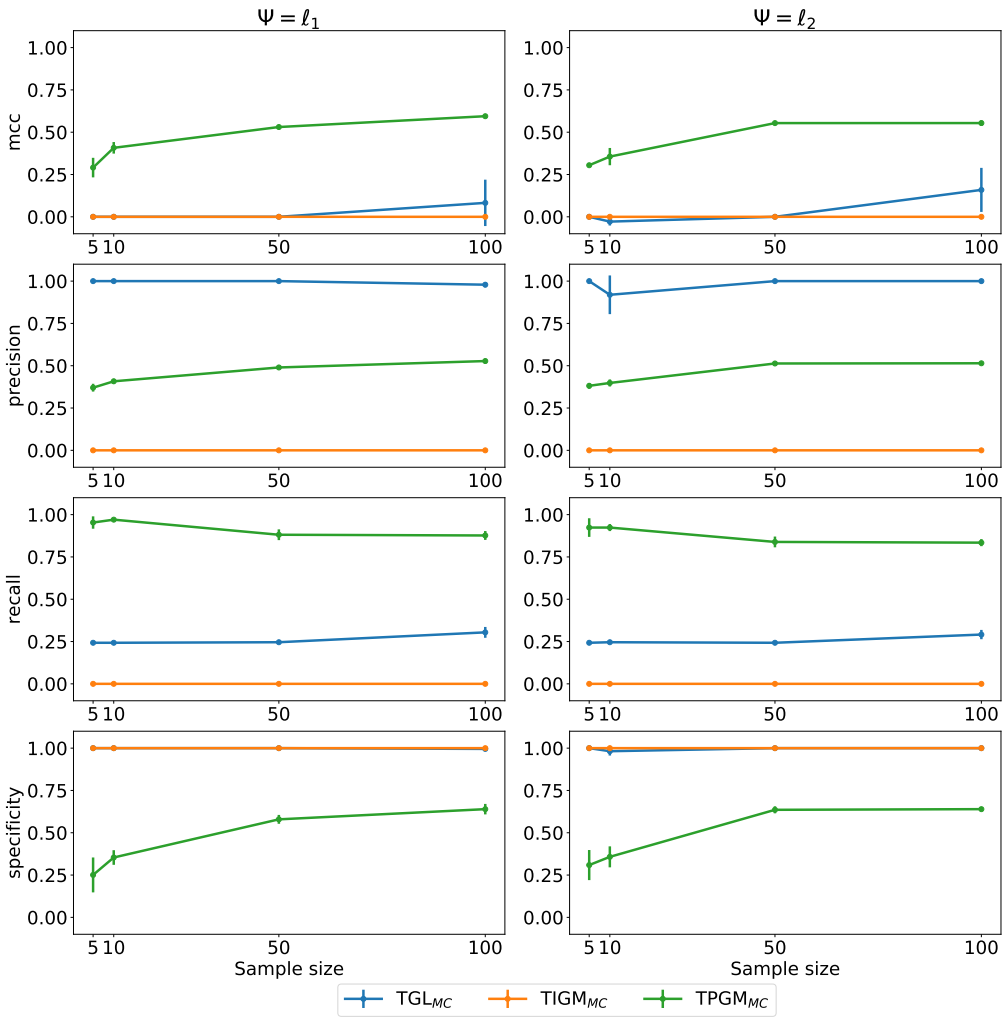


FIGURE 27. *Average results across 10 repetitions in terms of Matthew Correlation Coefficient (mcc, first row), precision (second row), recall (third row) and specificity (bottom row) for the comparison of Temporal Graphical Lasso with Multi-Class kernel (TGL_{MC}), Temporal Ising Graphical Model with Multi-Class kernel ($TIGM_{MC}$) and Temporal Poisson Graphical Model with Multi-Class kernel ($TPGM_{MC}$) at increasing sample size $N_t \in \{5, 10, 50, 100\}$ with $D = 10$ variables and 5 classes for multi-class experiments with scale-free random networks in case of ℓ_1 (first column) and ℓ_2 (second column) consistencies.*

5

Temporal graphical lasso with missing data

Part of this chapter content is present in the following publications:
Federico Tomasi, Veronica Tozzo*, Saverio Salzo and Alessandro Verri. Latent variable time-varying network inference. Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (2018). pp.2338- 2346*

Veronica Tozzo, Federico Tomasi, Margherita Squillario and Annalisa Barla. Group induced graphical lasso allows for discovery of molecular pathway-pathway interactions. Machine Learning for Health (ML4H) Workshop at NeurIPS (2018) - `arXiv181109673T`

Real-world observations often contain missing values. Consider the two following examples: during a medical trial patients subject to a survey refuse to answer some questions providing only partial information on their status; or, a genetic experiment measures genes through micro-arrays that provide information on only part of the whole genome. In the first example the resulting data matrix will have missing entries randomly positioned, while, in the second example, the data matrix will contain measurement of fewer variables than the one that are in play in the system. These two types of situations lead to two different concepts: in the first case, values are randomly missing and we call the corresponding variables *partial*, while, in the second case, values are missing with a pattern and the variables are *latent*.
These two types of missing data need to be analysed carefully and the related network inference methods should embed the missing data assumption. Indeed, ignoring the presence of missing values would lead to the inference of non-reliable graphs.
In literature, this problem has been tackled in the stationary case. In particular, (Städler and Bühlmann, 2012) and (Little and Rubin, 2019) considered partial data while (Anandkumar et al., 2013; Chandrasekaran, Parrilo and Willsky, 2010; Choi et al., 2011; Jalali et al., 2011; Yuan, 2012) considered latent data. All these methods assume data to be Gaussian is it allows to easily marginalise out the missing values. While the concept of missing data should be handled also for the other distribution, in this chapter we restrict to the Gaussian Graphical Models (GGMs) as well.

In particular, we take the temporal model presented in Section 4.3, and we study it under different conditions of missing data proposing extended models that take them naturally in consideration.

OUTLINE The rest of this chapter is organised as follows. Section 5.1 introduces the reader to the problems induced by missing data and the model for the inference of temporal GGMs from missing data. Section 5.2 describes in details the optimisation of such model based on the EM algorithm for both missing and latent data. Section 5.3 presents the convex alternative model for the inference of networks with latent variables. Section 5.4 illustrate a specialisation of the EM latent variable model that sees the latent variables as groups. Finally, Section 5.5 concludes with a discussion and future research directions.

5.1 Missing values in temporal models

The concept of missing values was introduced in Chapter 2, in particular we intend them as values that are un-observed and meaningful for a specific analysis. There, we also theoretically define the concept of data Missing at Random , *i.e.*, the mechanism that induces to un-observe data is ignorable. In addition we distinguished between two types of missing values:

- Partial: in absence of measurements randomly positioned in each sample;
- Latent: (or factored) in consistent absence of some variable measurements across all samples.

These two types of missing values introduce different problems during the inference of networks (Little and Rubin, 2019).

PARTIAL VARIABLES Partial data consist in a matrix X where missing values are randomly positioned with respect to both samples and variables. These holes in the matrix make impossible to directly perform computation on it without pre-processing or adopting ad-hoc inference mechanism. The pre-processing approaches could be the complete cases in which we discard the samples that do not have complete measurements on the variables or imputing using, as an example, the empirical mean. A visual representation of these two approaches can be seen in Figure 4. Note that, in the complete cases we reduce the sample size drastically which may impede the correct inference of the underlying graph especially when $N \ll D$.
On the other hand, imputing seems appealing as it induces to believe that we can reason in terms of complete data. Actually, imputing is dangerous as it introduces substantial bias in the estimated solution (Little and Rubin, 2019; Madow, Nisselson and Olkin, 1983).
Consider samples $X = (X[O], X[M])$ separated in observed and missing values. We could estimate the mean for each variables only on the observed part $\bar{X}[:, v] = \frac{1}{N} \sum_{i \in O_v} X[i, v]$ where with $\mathbb{I}_{O_v}$ we denote the set of indices of the samples that contain observations for the variable v. It can be shown that, the

sample variance obtained after imputing data with these means, has a rescaling factor of $\frac{|O|-1}{D-1}$ (Little and Rubin, 2019). Hence, imputation distorts the empirical distribution of the variables which consequently leads to bias in the estimate of the underlying graph. To solve this problem Städler and Bühlmann, 2012 proposed a EM algorithm in the stationary case that automatically estimated the partial value (see Section 2.3).

LATENT VARIABLES Latent variables can be seen as entities that are unobserved, thus we may not know their number nor the relationship they have with the observed variables. Their presence in the system though, if not taken into account leads to spurious edges, *i.e.*, links that would be conditioned away if the latent variables could have been observed (see Figure 6 for a visual description of the spurious edges) (Chandrasekaran, Parrilo and Willsky, 2010).
The presence of latent variables was tackled with different approaches in the stationary case: by fixing their number and possibly the structural relationship between latent and observed and use the EM algorithm to fit the parameters (Dempster, Laird and Rubin, 1977; Tozzo et al., 2018; Yuan, 2012). This approach has the draw-back of being non-convex, thus the optimisation could return local optima. Differently, Chandrasekaran, Parrilo and Willsky, 2010; Chandrasekaran et al., 2011 proposed a convex approach in which they estimate a graphical model on the observed variables and marginalise out the effect of the latent variables in order to delete the spurious edges.
The presence of missing values, both partial and latent, has never been tackled in the context of temporal models. In this chapter we address the problem of solving the Kernel Temporal Graphical Lasso (TGL_κ) problem in Equation (34) in presence of missing data. We argue that temporal consistency and dependency could improve network inference of dynamic networks as dependent time points would help with the absence of observations. We propose two possible minimisation approaches: the EM algorithm for partial and latent variables and the convex approach proposed by Chandrasekaran, Parrilo and Willsky, 2010 in the stationary case extended to the temporal case.

5.1.1 *Model*

At each time $t = 1, \dots, T$ we are given a matrix of observations $X_t \in \mathbb{R}^{N_t \times D}$ sampled from a multivariate normal distribution $\mathcal{N}(\mu_t, \Sigma_t)$. Such distribution is connected to a dynamical graphical model G through the precision matrices $K_t = \Sigma_t^{-1}$ that encode the conditional dependencies among variables. The observations of such variables may be non-complete, *i.e.*, some values may be missing in the matrix X_t. For each time t and sample i we denote with $\mathbb{I}_{O_i}$ and $\mathbb{I}_{M_i}$ the set of indices of observed and missing variables, respectively. Such sets allow us to divide the sample i in the following way

$$X_t[i,:] = \big(X_t[iO_i], X_t[iM_i]\big).$$

We want to remark that the sets O_i and M_i may change in time, so that sample i at time t may miss the observation of the variable v but it may have such observation at time $t+1$.
Then, under the hypothesis of incomplete observations, the goal is the inference of a dynamical GGM performed through the estimate of the precision matrices

$$\mathbf{K} = \left(K_1, \dots, K_T\right) \in \mathbb{R}^{(D\times D)\times T}$$

and the means

$$\boldsymbol{\mu} = (\mu_1, \dots, \mu_T) \in \mathbb{R}^{D\times T}.$$

The estimate of the means in this particular case is fundamental. Indeed, in presence of partial data it is impossible to assume zero mean as we did in previous chapters. This is due to the impossibility of computing the empirical mean and recentre the data without introducing bias. In fact, we would distort the empirical distribution in the same way imputation did.
The modelling and the inference of GGMs from these type of data is aid by the factorisation properties that hold for the multivariate normal distribution. In particular for each sample i it is possible to define a block precision matrices that groups the set of observed variables and the set of missing. From this grouping it is possible to obtain a conditional distribution that is still a multivariate normal distribution with parameters connected to the original one (Little and Rubin, 2019).
Given the sets O_i and M_i for every i we decompose each precision matrix K_t, for $t = 1, \dots, T$ as in Equation (24). Similarly we can decompose the mean vectors $\mu_t = (\mu_t[M_i], \mu_t[O_i])$.
With this separation, the inference problem can be defined as a MLE on the observed part of the data (see Theorem 1 in Section 2.1).
We recall to attention that we are always under the assumption of both temporal consistency and possibly non-Markovian temporal dependencies as we are extending the model presented in Section 4.3. Therefore, we assume matrices to be similar if they are close in time or if there is some complex variability pattern (*e.g.*, seasonality) present in the system (see 4.1 for a more thorough description).
Given a kernel κ that models temporal dependencies and a function Ψ that defines the type of temporal consistency, the functional can be written as

$$\begin{aligned}\underset{\substack{\boldsymbol{K},\boldsymbol{\mu}\\ K_t\succ 0}}{\text{minimise}} \sum_{t=1}^{T}\Bigg[\frac{1}{2}\sum_{i=1}^{N_t}\Big(\log\det\left(K_{tO_i}^{-1}\right) + \left(X_{tO_i} - \mu_{tO_i}^{\top}\right)\left(K_{tO_i}^{-1}\right)^{-1}\left(X_{tO_i} - \mu_{tO_i}\right)\Big) \\ + \alpha\|K_t\|_{od,1}\Bigg] + P_{\Psi,\kappa}(\boldsymbol{K})\end{aligned}$$

where $P_{\Psi,\kappa}(\boldsymbol{K})$ is the penalty defined in Equation (29).
It can be expressed in terms of sufficient statistics of the Normal distribution. At each time t we have two sufficient statistics: the sample means μ_t^C (Equa-

tion (21)) and the empirical covariance matrices $C_t = X_t^\top X_t$ (Equation (22)). The problem then becomes

$$\underset{\substack{\boldsymbol{K},\boldsymbol{\mu}\\ K_t \succ 0}}{\text{minimise}} \sum_{t=1}^{T} \Bigg[-\frac{N_t}{2}\log\det(K_t) + \frac{1}{2}\text{tr}(K_t C_t) + \frac{N_t}{2}\mu_t^\top K_t \mu_t + \mu_t^\top K_t \mu_t^C + \alpha \|K_t\|_{od,1} \Bigg] + P_{\Psi,\kappa}(\boldsymbol{K}) \tag{37}$$

Note that the functional that we just presented, under the assumption of complete data, is identical to the functional of Equation (34) plus the estimate of the mean. Indeed the loss could be rewritten as $-\frac{1}{2}\ell_{GGM}(K_t|X_t) + \frac{N_t}{2}\mu_t^\top K_t \mu_t + \mu_t^\top K_t \mu_t^C$. It also has a form consistent with the one in Equation (27), thus allowing for the use of all model selection techniques we introduced.

5.2 EM Algorithm

The model in Equation (37) allows for the inference of graphical models in case of both partial data, latent variables or a combination of the two (Little and Rubin, 2019; Städler and Bühlmann, 2012; Yuan, 2012) and can be optimised through the EM algorithm (see Section 2.2).
Problem (37) is non-convex, this may lead the related minimisation method to get stuck in local optima. For this reason we may require multiple initialisations in order to detect the final best reliable network.
We call such method *Kernel Missing Temporal Graphical Lasso* (MTGL$_\kappa$), and given its suitability for both partial and latent data, we call the related approaches MTGL$_\kappa^P$ and MTGL$_\kappa^L$ to differentiate. Note that the subscript κ entails the dependency from a specific kernel as in previous chapter, when we omit the subscript we are assuming a discrete kernel (see Equation (31)).

5.2.1 *Partial Data*

The EM algorithm for the complete data case is described in Algorithm 4. In particular the E-step is composed of two steps that computes the expectation of the sufficient statistics on the missing values. For each time t and each sample i the variables M_i are distributed according to a multivariate normal distribution

$$X_t[:, M_i]|X_t[:, O_i] \sim \mathcal{N}\Big(\mu_t[M_i] + K_t[M_i]^{-1}K_t[M_i O_i](X_t[:, O_i] - \mu_t[O_i]),\ K_t[M_i]^{-1}\Big)$$

Then, the missing values could be substituted with the mean of this conditional distribution. In particular we have that

$$\mathbb{E}[X_t[iv]|X_t[O_i], \mu_t^{\iota-1}K_t^{\iota-1}] = \begin{cases} X_t[iv] & \text{if variable } v \text{ is observed for sample } i \\ c_{ti}[v] & \text{if is missing} \end{cases}$$

Algorithm 4 EM algorithm for MTGL^P

Inputs: Ψ consistency function, κ temporal dependencies, $\boldsymbol{X}$ samples, α sparsity hyper-parameters
for $\iota = 1, \dots,$ **do**
  //E-step
  for $t = 1, \dots T$ **do**
    $\mathbb{E}[X_t^\iota[iv]|X_t[:, O_i], \mu_t^{\iota-1} K_t^{\iota-1}]$
    $\mathbb{E}[C_t^\iota|X_t[:, O_i], \mu_t^{\iota-1} K_t^{\iota-1}]$
  //M-step
  for $t = 1, \dots T$ **do**
    $\mu_t^\iota = \frac{1}{N_t}(\sum_{i=1}^{N_t} X_{i1}, \dots, \sum_{i=1}^{N_t} X_{iD})$
  $\boldsymbol{K}^\iota = \underset{\boldsymbol{K} \succ 0}{\operatorname{argmin}} \sum_{t=1}^{T} [-N_t \ell_{GGM}(C_t^\iota|K_t) + \alpha \|K_t\|] + P_{\Psi,\kappa}(\boldsymbol{K})$

Algorithm 5 EM algorithm for MTGL^L

Inputs: Ψ consistency function, κ temporal dependencies, $\boldsymbol{X}$ samples, α sparsity hyper-parameters
for $\iota = 1, \dots,$ **do**
  //E-step
  for $t = 1, \dots T$ **do**
    $\mathbb{E}[C_t^\iota|X_t[O], \mu_t^{\iota-1} K_t^{\iota-1}]$
  //M-step
  $\boldsymbol{K}^\iota = \underset{\boldsymbol{K} \succ 0}{\operatorname{argmin}} \sum_{t=1}^{T} [-N_t \ell_{GGM}(C_t^\iota|K_t) + \alpha \|K_t\|] + P_{\Psi,\kappa}(\boldsymbol{K})$

where $c_{ti}[j]$ is the $j-th$ entry of the vector $c_{ti} \in \mathbb{R}^{|M_i|}$ defined as the mean of the conditional distribution

$$c_{ti} = \mu_t[M_i] + K_t[M_i]^{-1} K_t[M_i O_i](X_t[:, O_i] - \mu_t[O_i]) \tag{38}$$

The computation of the expectation of the empirical covariance is computed similarly, exploiting the mean and computing

$$\mathbb{E}[X[iv]X[iv']|X_t[O_i], \mu_t^{\iota-1} K_t^{\iota-1}] = \begin{cases} X_t[iv]X_t[iv'] & \text{if } v, v', \text{ observed} \\ X_t[iv]c_{ti}[v'] & \text{if } v \text{ observed} \\ (K_t[M_i]^{-1})_{vv'} + c_{ti}[v]c_{ti}[v'] & \text{otherwise} \end{cases}$$

where c_{ti} is computed as in Equation (38).
Then the entry vv' of the covariance matrix is computed as

$$C_t^\iota[vv'] = \sum_{i=1}^{N_t} \mathbb{E}[X[iv]X[iv']|X_t[O_i], \mu_t^{\iota-1} K_t^{\iota-1}]$$

5.2.2 *Latent Data*

Dealing with latent data allows us to take two assumptions that simplify the inference process.

1. The mean is zero for all variables: such assumption can be taken in the case of latent variables as we are not introducing any bias given the presence of all values for the observed variables.

2. The missing values are always the same for all the samples: this is implied by latent variables definition. Indeed, for all the samples and for all the time points, such variables are un-observed and therefore latent. This also implies that, given the fact that we cannot observe them, we may also not know how many they are.

Consider now the model presented in Equation (37), in a real system formed by latent variables and observable variables we can define two sets of indices as $\mathbb{I}_M = 1, \dots, M$ and $\mathbb{I}_O = M+1, \dots, D$ for the latent and observed variables respectively. Given these two sets we can define a simpler model as follows:

$$\begin{aligned} \underset{\boldsymbol{K}, K_t \succ 0}{\text{minimise}} \sum_{t=1}^{T} & -\ell_{GGM}(X_t[:\mathrm{O}]|K_t[O]) + \alpha \|K_t\|_{od,1} \\ & + P_{\Psi,\kappa}(\boldsymbol{K}) \qquad \text{s.t.} |M| = r. \end{aligned} \tag{39}$$

Note that the constraint $|M| = r$ states that the cardinality of the set $\mathbb{I}_M$ should be r, *i.e.*, we have r latent variables. The newly introduced hyper-parameter r can be imposed if prior knowledge is available or identified via model selection strategies (see Chapter 3 for details on the possible methods). When knowledge on the number of latent variables, or their identity, is known it could be possible to further guide the inference. We study such case later in Section 5.4.
The algorithm for the minimisation of the functional is described in Algorithm 5. Differently from Algorithm 4, it requires less step as we do not need to estimate the mean. Moreover, the computation of the expectation of the covariance matrix can be performed in blocks.
We define the matrix c_t as

$$c_t = K_t[M]^{-1} K_t[MO] X_t[O]$$

then the expectation is computed as

$$\mathbb{E}[C_t^\iota | X_t[O], \mu_t^{\iota-1} K_t^{\iota-1}] = X_t^\top X_t = \begin{bmatrix} K_t[M]^{-1} + c_t c_t \top & c_t X_t[O]^\top \\ X_t[O] c_t^\top & X_t[O]^\top X_t[O] \end{bmatrix}$$

Note that, by optimising Problem (39) we obtain an estimate of the precision matrix also in the part corresponding to the latent variables $K_t[M]$

5.2.3 *Synthetic Data Experiments*

We need to assess the reliability of both MTGL_κ^P and MTGL_κ^L. Given the latent nature of the second, though, we post-pone the narration of its synthetic experiments to next section to include also the model that marginalise the latent effect.

We devised one experiment that puts in comparison MTGL^P_κ with the Kernel Temporal Graphical Lasso (TGL_κ), both with an RBF kernel. We generate $T = 10$ temporal precision matrices letting them evolve with an ℓ_1 behaviour (see Appendix C.1). From each distribution given by the precision matrices we sample $N_T = 100$ samples for $D = \{20, 100\}$ variables. We apply TGL_κ on such data and then we randomly inserted some missing values with percentage $\{10, 20, 30\}$ of the total number of available samples. Higher percentages would lead the algorithm to non-convergence. We selected the hyperparameters using Bayesian search optimisation (Molinaro, Simon and Pfeiffer, 2005). We repeated data generation and inference 10 times to assess the reliability of the method.

5.2.4 *Results*

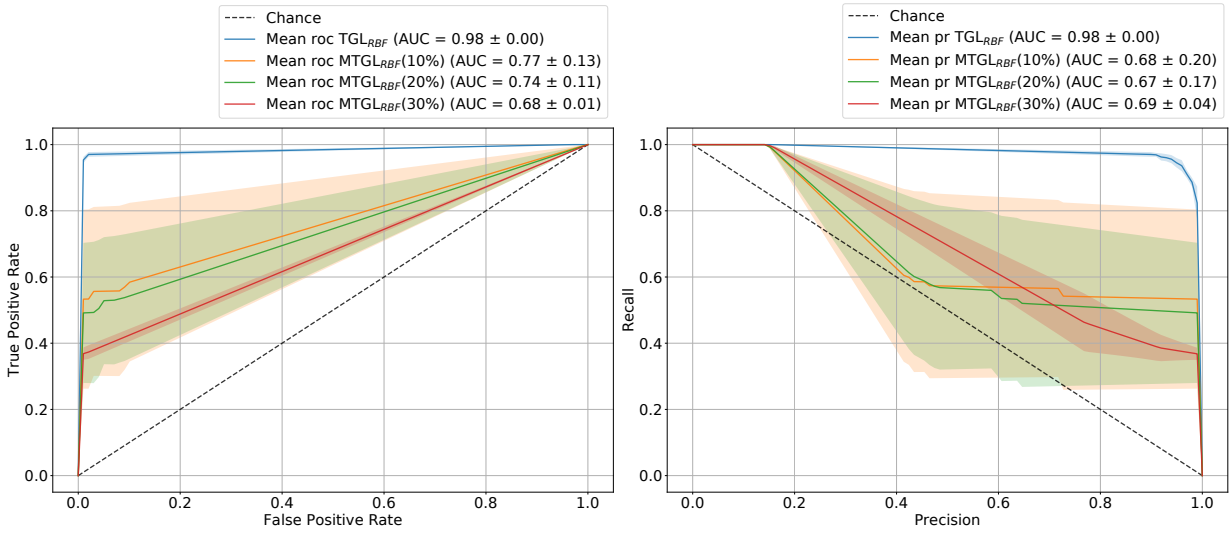
TGL_{RBF} has excellent performance in retrieving the dynamic network while MTGL^P_{RBF} is able to retrieve the network under an increasing percentage of partial values with good approximation.

Figure 28 and Figure 29 show the ROC and Precision-Recall curve obtained for $D = 20$ and $D = 100$ respectively. TGL_{RBF} performs consistently better than MTGL^P_{RBF} when the percentage of missing data increases. Nonetheless, MTGL^P_{RBF} performances are above chance and remain consistent as the number of available values decrease. Interestingly, in both experiments we observe that for a percentage of 30% of partial values the algorithm has higher AUC than with other inferior percentages. Nonetheless, if we observe Table 6 where we present the scores for the algorithms, we note that MTGL^P_{RBF} has a consistently decrease in scores as the number of missing values increases. We want to remark that the 10 repetitions were performed by randomly generating a new dataset each time. Therefore, possible re-initialisation of the algorithm would lead to better performances.

A comparison in terms of time to converge of the two methods is present in Chapter 6 (Figure 33 and Figure 34).

	$D = 20$				$D = 100$			
	TGL	(10%)	(20%)	(30%)	TGL	(10%)	(20%)	(30%)
P	$\mathbf{0.68 \pm 0.02}$	0.40 ± 0.41	0.40 ± 0.42	0.5 ± 0.41	$\mathbf{0.35 \pm 0.41}$	0.01 ± 0.01	0.02 ± 0.01	0.01 ± 0.01
R	$\mathbf{0.96 \pm 0.01}$	0.36 ± 0.36	0.29 ± 0.29	0.05 ± 0.04	$\mathbf{0.69 \pm 0.39}$	0.28 ± 0.38	0.22 ± 0.30	0.21 ± 0.30
F_1	$\mathbf{0.80 \pm 0.01}$	0.38 ± 0.39	0.33 ± 0.34	0.09 ± 0.07	$\mathbf{0.15 \pm 0.08}$	0.02 ± 0.02	0.02 ± 0.01	0.01 ± 0.02
S	0.95 ± 0.00	0.93 ± 0.05	0.95 ± 0.03	$\mathbf{0.99 \pm 0.01}$	0.70 ± 0.36	0.73 ± 0.37	0.77 ± 0.31	$\mathbf{0.80 \pm 0.28}$
BA	$\mathbf{0.95 \pm 0.00}$	0.65 ± 0.20	0.62 ± 0.16	0.52 ± 0.02	$\mathbf{0.69 \pm 0.16}$	0.50 ± 0.01	0.50 ± 0.01	0.50 ± 0.01

TABLE 6. *Average performance across 10 repetitions in terms of Precision (P), Recall (R), F_1-score (F_1), Specificity (S) and Balanced Accuracy (BA) for the comparison of Temporal Graphical Lasso (TGL_{RBF}) with the Missing Temporal Graphical Lasso with Partial data at different percentages of missing values for two networks of $D = \{20, 100\}$ nodes, $N_t = 100$ samples and $T = 10$ times.*



(A) *ROC curve 20 nodes* (B) *Precision-Recall curve 20nodes*

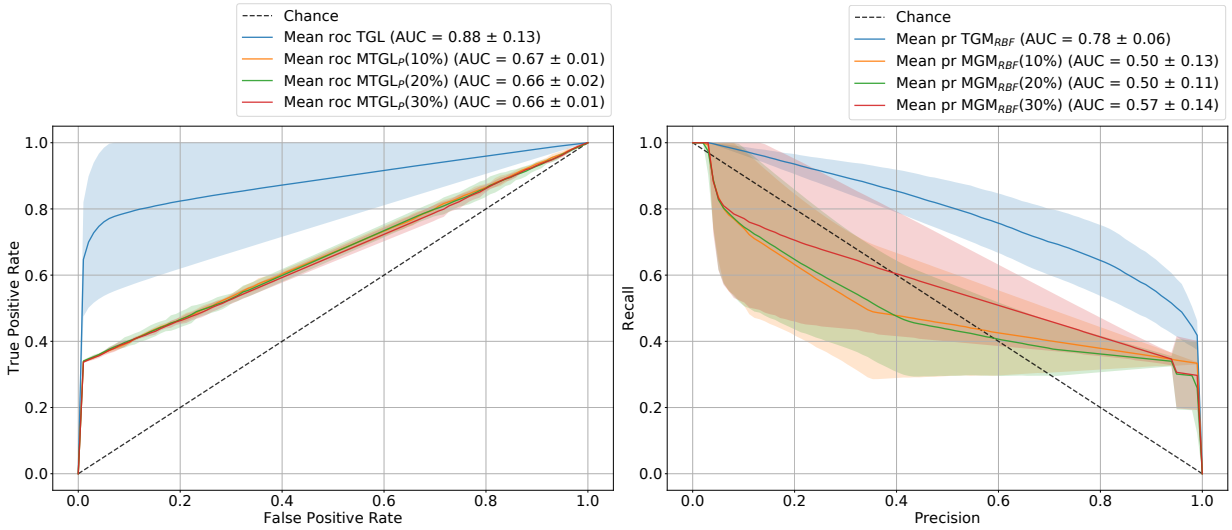
FIGURE 28. *Average results across 10 repetitions in terms of ROC and PR curves for the comparison of Temporal Graphical Lasso with RBF kernel (TGL$_{RBF}$) and Missing Temporal Graphical Lasso for Partial data with RBF kernel MTGL$^{P}_{RBF}$ for the inference of a network on $D = 20$ nodes, with $N_t = 100$ samples and $T = 10$ time points at an increasing number of missing values* $\{10, 20, 30\}$.

5.3 Latent Variables Marginalisation

The MTGL$_\kappa$ model is non-convex. In case of latent variables we can overcome this problem by optimising a different but related method that 1. is convex thus is guaranteed to converge to a global optimum; 2. allows to decouple the temporal behaviour between latent and observe variables allowing for an increase in expressive power. Such method, that we call *Kernel Latent Variable Temporal Graphical Lasso* (LTGL$_\kappa$), does not directly estimate the latent variables but learns their effect allowing for it to be marginalised out of the final solution. Consider the model in Equation (39), the r latent variables that are not measured lead to perturbed observations. If we do not consider the entire multi-variate distribution but only the observed portion this will have another distribution that strictly depends on the complete one. In particular we would have samples from a perturbed dynamical graphical model as follows

$$(X_1, \dots, X_T) \sim (\mathcal{N}(0, \tilde{\Sigma})_1, \dots, \mathcal{N}(0, \tilde{\Sigma})_T)$$

where, $\tilde{\Sigma}_i = \tilde{K}_t[O]^{-1}$ and, for each $t = 1, \dots, T$, the perturbed observed precision matrix is defined as the Schur complement in Equation (25).
This idea was presented in (Chandrasekaran, Parrilo and Willsky, 2010) in the stationary case (see Section 2.4) and in (Foti et al., 2016) for the analysis of MEG time-series. The goal is the inference of both $K_t[O]$ and L_t simultaneously in such a way that the matrix L_t contains the effect of the latent variables on the system and allows it to be marginalised out. Note that, by definition, the matrices L_t have rank r that corresponds to the number of latent variable $|M|$,



(A) *ROC curve 20 nodes* (B) *Precision-Recall curve 20nodes*

FIGURE 29. *Average results across 10 repetitions in terms of ROC and PR curves for the comparison of Temporal Graphical Lasso with RBF kernel (TGL$_{RBF}$) and Missing Temporal Graphical Lasso for Partial data with RBF kernel MTGL$^P_{RBF}$ for the inference of a network on $D = 100$ nodes, with $N_t = 100$ samples and $T = 10$ time points at an increasing number of missing values* $\{10, 20, 30\}$.

therefore, similarly to the constraint $|M| = r$ in Equation (39) we need to impose such value during the optimisation.
Ideally, we would impose the constraint $\text{rank}(L_t) = r$, which, in turn, would lead to a non-convex problem. Nonetheless, we can relax it through the nuclear norm, in Equation (26) that keeps the problem convex while still retaining guarantees of retrieving the true underlying model (Chandrasekaran et al., 2011)
Chandrasekaran et al., 2011 showed that given this model, for the correct identification of the terms $K_t[O]$ and L_t we require two strict assumption on the data. In particular we have that the number of latent variables r must be low respect to the number of observed and their effect must be spread out on the observed variables. This second assumptions translates in the fact that the latent variables are connected with the majority of the observed variables (see Section 2.4.2)
The inference is then aimed at inferring a set of sparse matrices $\boldsymbol{K}[O] = (K_1[O], \dots, K_T[O])$ and a set of low-rank matrices $\boldsymbol{L} = (L_1, \dots, L_T)$ such that, at each time point t, $K_t[O]$ encodes the conditional independences between the observed variables, while L_t provides the summary of marginalisation over latent variables on the observed ones. Similarly to the other models we want to impose temporal consistency and dependency. The peculiarity of the decoupling between latent and observed parts is that we can also decouple the types of temporal consistency we impose on the observed and latent part of model allowing for more expression power. We use two consistency function: Ψ that acts on the observe part of the network and Φ that acts on the latent

marginalisation. The temporal dependency are instead specify by the kernel κ. The *Kernel Latent-variable Time-varying Graphical Lasso* (LTGL$_\kappa$) model takes the following form:

$$\begin{aligned}\underset{K_t \in \mathcal{S}^D_{++}, L_t \in \mathcal{S}^D_{+}}{\text{minimise}} \sum_{t=1}^{T} \Big[- \ \ell_{GGM}(X_t | K_t - L_t) + \alpha \|K_t\|_{1,od} + \tau \|L_t\|_* \Big] \\ + P_{\Psi,\kappa}(\boldsymbol{K}) + P_{\Phi,\kappa}(\boldsymbol{L}).\end{aligned} \tag{40}$$

The main advantage of this model is its convexity that guarantees to reach the global optimum. Indeed, under the assumptions of low-rank and spread influence of the latent variables effect, the model converges to a solution that is accurate with high probability (Chandrasekaran et al., 2011). It also permit more flexibility in the imposition of temporal consistency allowing for complex patterns retrieval.

5.3.1 *Minimisation Algorithm and Automatic Kernel Discovery*

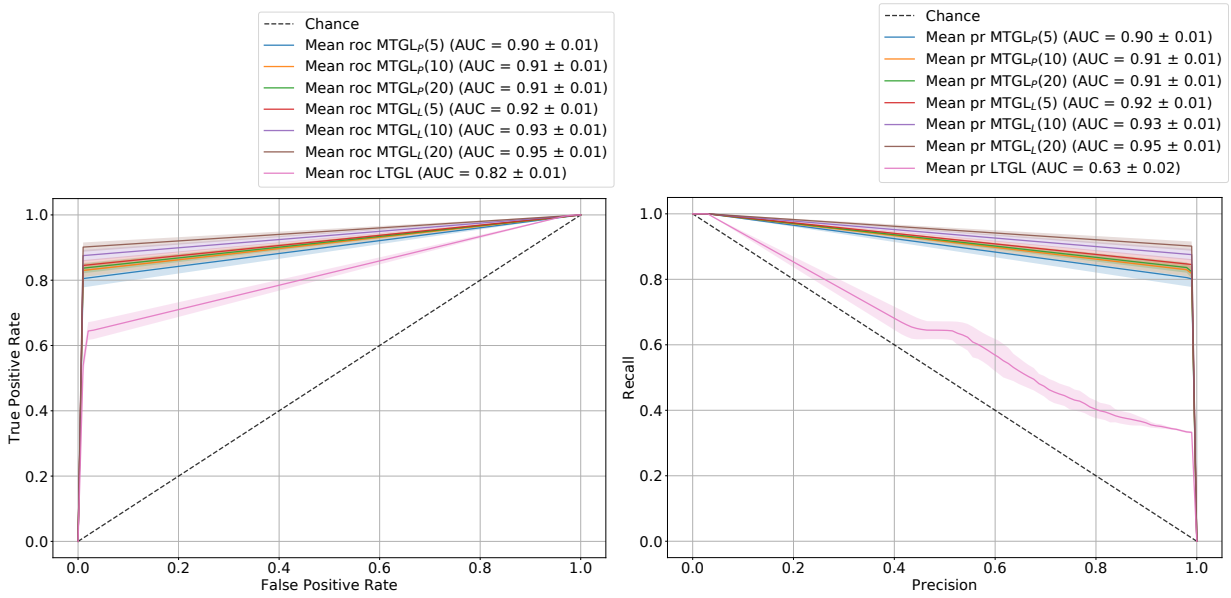
The minimisation algorithm of LTGL$_\kappa$, with fixed kernel, is based on the alternating direction method of multipliers (ADMM) (Boyd et al., 2011). The derivation of such algorithm is non-trivial, therefore, to improve readability we provide its description in Appendix B.
Nonetheless, as for the generalised temporal models, the knowledge on the kernel κ may be limited. Therefore, we can recur to automatic identification of temporal dependencies (see Section 4.2). We propose such extension only in the case of the LTGL$_\kappa$ model as it has guarantees of always reaching the global optimum. Indeed, MTGL$_\kappa$ already entails a non-convex minimisation approach and inserting it in a further optimisation procedure may lead to non reliable results. The minimisation procedure for the automatic inference of the kernel is the same proposed in Section 4.2 and we call this variation *Latent-variables Time-varying Graphical Lasso with Pattern discovery* (LTGL$_P$).

5.3.2 *Synthetic Data Experiments*

We performed experiments on synthetic data assessing the performance of MTGL$^L_\kappa$ and LTGL$_\kappa$ in terms of recovery of the structure on the observed and latent part of the graph. The hyper-parameters were selected with a Bayesian optimisation procedure based on expected improvement strategy (Snoek, Larochelle and Adams, 2012) through a 3-fold cross-validation.

MISSING-DATA METHODS COMPARISON

In this experiment we wanted to assess the goodness of MTGL$^P_{RBF}$, MTGL$^L_{RBF}$ and LTGL$_{RBF}$. We generated a dataset with ℓ_1 evolving behaviour (see Appendix C.1). The observed variables were $|O| = 100$ and the latent $|M| = 5$ for a total of $D = 105$ variables. We sampled $N_t = 100$ observations at each time $t = 1, \ldots, 10$ from the perturbed distribution. In order to apply MTGL^P



(A) *ROC curves* (B) *Precision-Recall curves*

FIGURE 30. *Average results across 10 repetitions in terms of ROC and PR curves for the comparison of the Latent Temporal Graphical Lasso (LTGL), the Missing Temporal Graphical Lasso for Partial data (MTGL$_P$) and the Missing Temporal Graphical Lasso for Latent data (MTGL$_L$) for the inference of a network on $D = 105$ nodes (100 observed and 5 latent), with $N_t = 100$ samples and $T = 10$ time points. The MTGL methods were applied with different instantiation of the hyper-parameter r (number in round brackets) that sets the number of latent variables.*

we add r columns on the input matrix containing NaN values. All the hyper-parameters were cross-validated except the hyper-parameter r corresponding to the number of latent variables in MTGL^P and MTGL^L that we fixed a priori. We applied the methods with an increasing value of r to see if the methods were able to infer correctly the precision matrix disregarding the number of latent variables. We repeated the experiments 10 times to assess the reliability of the methods in presence of different datasets.

COMPARISON OF CONVEX METHODS

We compare LTGL with discrete kernel with the Graphical Lasso (GL) (Friedman, Hastie and Tibshirani, 2008), the Latent Variable Graphical Lasso (LGL) (Chandrasekaran, Parrilo and Willsky, 2010; Ma, Xue and Zou, 2013) and Time-Varying Graphical Lasso (TVGL) (Hallac et al., 2017a). We devised two types of temporal consistency: ℓ_2^2 perturbation (p_2) (Appendix C.2) and ℓ_1 perturbation (p_1) (Appendix C.1). For (p_2) we generated a dataset with $|O| = 100$, $|M| = 20$, $T = 10$ and $N_t = 100$ samples. For this reason, in this setting, the contribution of latent factors is predominant with respect to the network evolution in time. For (p_1) we generated a dataset with $|O| = 50$, $|M| = 5$, $T = 10$ and $N_t = 100$ samples. In this setting, the temporal component affects the network more than the latent factor contribution.

perturbation	method	score			
		F_1	ACC	MRE	MSE
ℓ_2^2 (p_2)	LTGL (ℓ_2^2)	**0.926**	**0.994**	**0.70**	0.007
	LTGL (ℓ_1)	0.898	0.993	**0.70**	0.007
	TVGL (ℓ_2^2)	0.791	0.980	-	**0.003**
	TVGL (ℓ_1)	0.791	0.980	-	**0.003**
	LVGLASSO	0.815	0.988	2.80	0.007
	GL	0.745	0.974	-	0.004
ℓ_1 (p_1)	LTGL (ℓ_2^2)	0.842	0.974	0.29	0.013
	LTGL (ℓ_1)	**0.880**	**0.981**	**0.28**	0.013
	TVGL (ℓ_2^2)	0.742	0.950	-	0.009
	TVGL (ℓ_1)	0.817	0.968	-	0.009
	LVGLASSP	0.752	0.964	0.74	0.013
	GL	0.748	0.951	-	**0.007**

TABLE 7. *Performance in terms of F_1-score (F_1), Accuracy (ACC), Mean Rank Error (MRE) and Mean Squared Error (MSE) for the comparison of Latent Temporal Graphical Lasso with discrete kernel (LTGL), the Time-Varying Graphical Lasso (TVGL), the Latent Variable Graphical Lasso (LVGLASSO) and the Graphical Lasso (GL) for two different types of evolutionary patterns (ℓ_1 and ℓ_2^2)*

NON-MARKOVIANITY

We compare LTGL with discrete kernel, LTGL$_{ESS}$ and $LTGL_P$ with automatic pattern discovery. Given $T = 20$ times in $D = 100$ and $N_t \in \{5, 10, 50, 100, 500\}$ we generated data according to the temporal conditioning schema (see Appendix C.5) with evolution decided with a cluster-based generation (see Appendix C.4). The latent factor, in this case, is given by the conditioning on the previous time stamp.

5.3.3 *Results*

MISSING-DATA METHODS COMPARISON

MTGL^P and MTGL^L have similar results also with different numbers of estimated latent variables and, both, outperform LTGL.

Figure 30 shows the Precision-Recall and ROC curves of the comparison between MTGL^P, MTGL^L and LTGL. The results show that MTGL^P and MTGL^L perform similarly, which is in line with the fact that, except for the mean estimation, they are identical methods. Note that the difference in the imposed number of latent variable does not impact significantly on the estimate of the precision matrix. This means that the selection of a further hyper-parameter is not a major concern in the use of these two methods. On the other hand, LTGL

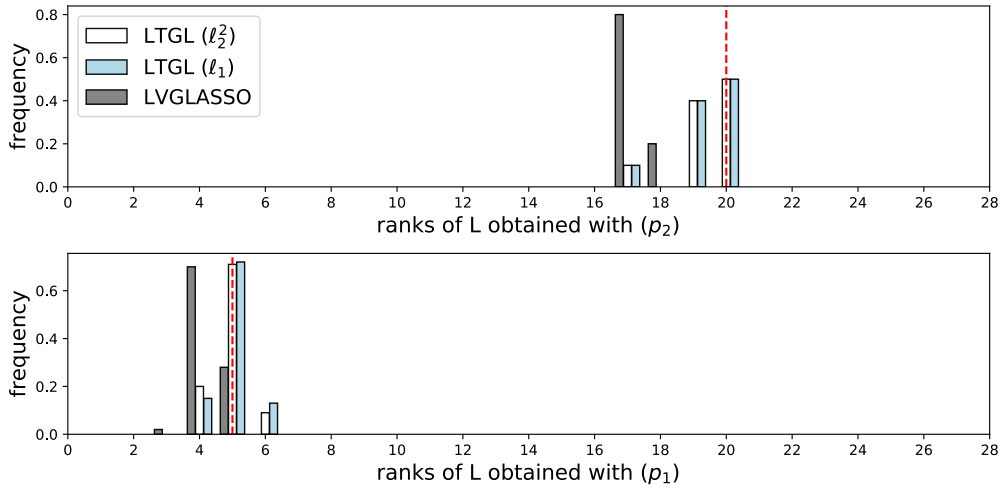


FIGURE 31. *Distribution of inferred ranks across all time points for the Latent Temporal Graphical Lasso (LTGL) and the Latent Variable Graphical Lasso (LVGLASSO). The vertical line indicates the ground truth rank, around which all detected ranks lie. Note that, in* (p_2), $L_t \in \mathbb{R}^{100\times100}$, *therefore the range of possible ranks is* $[0, 100]$. *For* (p_2), $L_t \in \mathbb{R}^{50\times50}$, *hence the range is* $[0, 50]$.

performs poorly. We argue that this could be, in fact, due to the tuning of its four hyper-parameters, especially τ that controls the number of latent variables. We noted that the combination of τ and α is non-stable and it requires further theoretical investigation.

COMPARISON OF CONVEX METHODS

LTGL outperforms all convex state-of-the-art methods under temporal and latent variable data. Table 7 shows the performance of LTGL compared with the other methods for both types of perturbation ℓ_2^2 (p_2) and ℓ_1 (p_1). Note that MRE is not available for all the methods since neither GL or TVGL consider latent factors. LTGL and TVGL are used with two temporal penalties according to the different perturbation models of data generation. In this way, we show how the correct choice of the penalty for the problem at hand results in a more accurate network estimation. In both (p_2) and (p_1), LTGL outperforms the other methods for graphical modelling. In (p_2), in particular, LTGL correctly infers almost 99,5% of edges in all the dynamical network both with the ℓ_2^2 and ℓ_1 penalties. Nonetheless, the use of ℓ_2^2 penalty enhance the quality of the inference as expected from the theoretical assumption made during data generation. Both the choice of a penalty that reflects the way in which data are generated and time consistency are reflected in a low MRE, which encompasses LGL ability in detecting latent factors (Figure 31). In (p_2), in fact, the number of latent variables with respect to both observed variables and samples is high. Therefore, by exploiting temporal consistency of the network, LTGL is able to improve the latent factors estimation. Simultaneous consideration of time and latent variable also positively influences the F_1 score. Above considerations also hold for the (p_1) setting. Here, LTGL achieves the best results in both F_1 score and accuracy, while having a low MRE. The adoption of

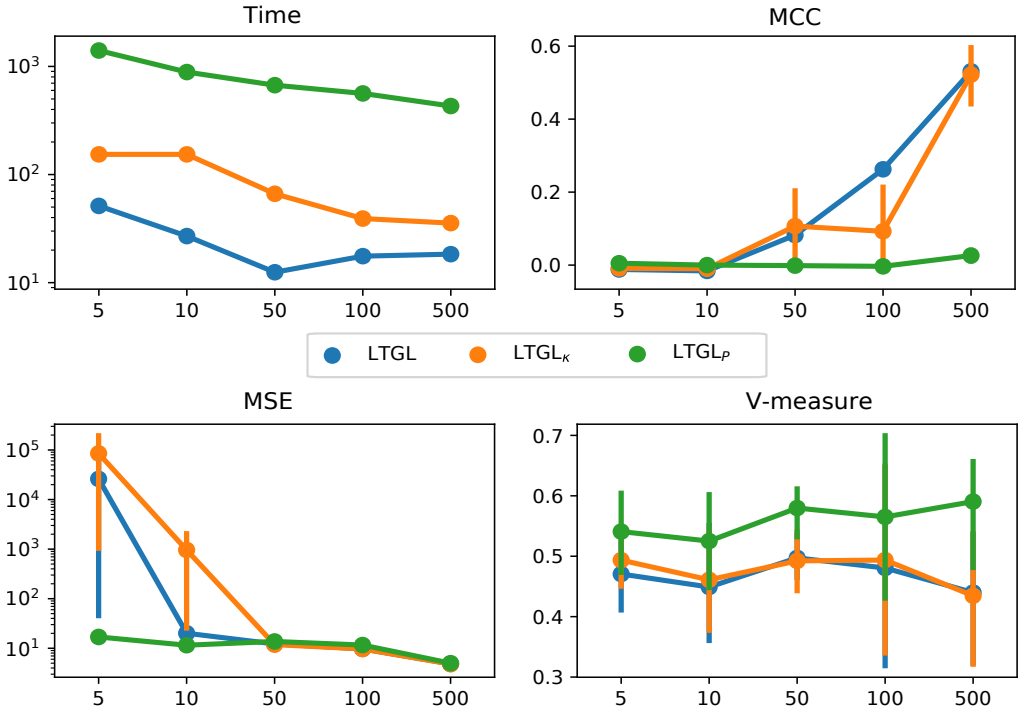


FIGURE 32. *Performance of Time (in seconds), Matthew Correlation Coefficient (MCC), Mean Squared Error (MSE) and V-measure for the Latent Temporal Graphical Lasso, with ESS kernel (LTGL$_\kappa$), with automatic Pattern inference (LTGL$_P$) and with discrete kernel (LTGL) for one experiment on complex temporal dependencies on a network of $D = 100$ dimensions, $T = 20$ times and increasing sample size $N_t \in \{5, 10, 50, 100, 500\}$.*

ℓ_1 penalty improves structure estimation and latent factors detection, consistently with the data generation model. Such settings were designed to show how the prevalence of latent factors contribution or time consistency affects the outcome of a network inference method. In (p_2), where the latent factors contribution is prevalent, network inference is more precise when considering latent factors. In (p_1), instead, the number of time points is more relevant than the contribution of latent factors, hence it is more effective to exploit time consistency (both for latent and observed variables), evident from the results of Table 7. LTGL benefits from both aspects.

NON-MARKOVIANITY

In presence of complex temporal dependency the use of kernels improves structure identification while pattern detection infers the correct clusters with high accuracy.

Figure 32 depicts the trend of scores as the number of samples increases keeping fixed the number of dimensions. Results show that, as in Chapter 4, considering a temporal kernel is beneficial to better approximate the evolution of the system under analysis. Indeed, LTGL$_{ESS}$ and LTGL$_P$ perform better than LTGL with discrete kernel especially when $N \ll D$. LTGL$_P$ require more time to converge given the two-steps alternating minimisation procedure. Table 8 shows a detailed performance when N_t is fixed to 50. Here, LTGL$_{ESS}$ has higher performance than LTGL$_P$ that, in turns, shows a better V-measure. The imposition

	LTGL	LTGL$_\kappa$	LTGL$_P$
BA	0.509 ± 0.002	$\mathbf{0.521 \pm 0.016}$	0.500 ± 0.000
P	0.278 ± 0.003	$\mathbf{0.299 \pm 0.045}$	0.251 ± 0.002
MCC	0.082 ± 0.010	$\mathbf{0.107 \pm 0.117}$	-0.001 ± 0.002
MSE	12.229 ± 0.279	$\mathbf{11.861 \pm 0.483}$	13.711 ± 0.117
V-measure	0.497 ± 0.037	0.492 ± 0.058	$\mathbf{0.580 \pm 0.047}$

TABLE 8. *Average performance across 10 repetitions in terms of balanced accuracy (BA), precision (P), Matthews correlation coefficient (MCC), mean squared error (MSE) and V-measure for the comparison of the Latent Temporal Graphical Lasso, with ESS kernel (LTGL$_\kappa$), with automatic Pattern inference (LTGL$_P$) and with discrete kernel (LTGL) on a dynamical network of $T = 20$ time points and $D = 100$ variables at sample size $N_t = 50$.*

of the *ESS* kernel drives the model to outperform the competitors in terms of balanced accuracy, average precision and MCC.

5.4 Prior on latent variables identity

The MTGL^L model allows to obtain an estimate on the latent part of the precision matrix. Here, we want to exploit the presence of prior knowledge to guide the inference towards multi-layer inference of networks. Indeed, under the assumption that latent variables should be few with respect to the observed and connected to a great number of them we could think of a latent variable as a group of observed ones. For example, in biological contexts one may want to "marginalise" out from the graph the effect of groups of genes, where different groups (latent variables) may lead to changes in the observed part of the network. Groups in this case may be pathways, biological processes, molecular functions and others (Tozzo et al., 2018). We call such method Missing Temporal Graphical lasso with Group imposition (MTGL$_G$)
Such approach is similar to the method proposed by (Cheng, Shan and Kim, 2017) where a group-lasso penalty on the network groups genes within pathways. This approach, nonetheless, forbid links from genes belonging to different pathways to be inferred. In other words they assume the pathway-pathway interactions to completely explain the dynamics of the system. We argue that this may be reductive in practice where more complex connections may be in play.

Consider N_t observations on D variables where the set $\mathbb{I}_O$ is the set of observed ones. We can only observe $X_t \in \mathbb{R}^{N_t \times O}$ drawn from a multivariate Gaussian distribution $\mathcal{N}(0, \tilde{\Sigma}_t[O])$, where $\tilde{\Sigma}_t[O]$ is a perturbed covariance matrix whose inverse $K_t[O]$ has the same form as in Equation (25). Here, the variables indexed with the set $\mathbb{I}_M$ are not simply latent but are assumed to be groups of observed variables. We can codify the membership of each observe variables

(*e.g.*, gene) to a specific group (*e.g.*pathway) in a binary matrix $G \in \{0,1\}^{|O|\times|M|}$ where $G_{om} = 1$ if the observed variable o belongs to the group m and 0 otherwise. Note that groups can overlap but they are stable in time, indeed we do not expect for example genes to belong to different pathways as time passes. Also, we do not expect such memberships to exhaustively explain all links between observed variables. In other words, we do not assume the system to be completely explained by the un-observed variables. Our goal is to estimate the precision matrices $\boldsymbol{K} = (K_1, \dots, K_T)$ of the form Equation (24) where $K_t[M]$ and $K_t[O]$ represent the precision matrices between groups and single variables respectively. Note that in the inferred $K_t[OM]$ sub-matrix a non-zero entry should be found in correspondence of non-zero entries of G. The prior knowledge on the groups can be easily imposed in the model (39) by using a matrix penaliser instead of the parameter α as follows:

$$\begin{aligned}\underset{\boldsymbol{K},K_t\succ 0}{\text{minimise}} \sum_{t=1}^{T} -\ell_{GGM}(X_t[:\mathrm{O}]|K_t[O]) + \|A \odot K_t\|_{od,1} \\ + P_{\Psi,\kappa}(\boldsymbol{K}) \text{ s.t.} |M| = r\end{aligned} \tag{41}$$

where $\odot$ denotes the element-wise product between matrices and A is defined as

$$A = \left[\begin{array}{ccc|ccc} 0 & \dots & \alpha & \mu\bar{G}_{11} & \dots & \mu\bar{G}_{O1} \\ \alpha & \dots & \alpha & \mu\bar{G}_{12} & \dots & \mu\bar{G}_{O2} \\ \vdots & & \vdots & \vdots & \ddots & \vdots \\ \alpha & \dots & 0 & \mu\bar{G}_{1M} & \dots & \mu\bar{G}_{OM} \\ \hline \mu\bar{G}_{11} & \dots & \mu\bar{G}_{1H} & 0 & \dots & \alpha \\ \mu\bar{G}_{21} & \dots & \mu\bar{G}_{2H} & \alpha & \dots & \alpha \\ \vdots & & \ddots & \vdots & \ddots & \\ \mu\bar{G}_{O1} & \dots & \mu\bar{G}_{OH} & \alpha & \dots & 0 \end{array}\right]. \tag{42}$$

Here, the value μ is needed to determine how strongly the structured regularisation $\bar{G} = 1 - G$ is enforced on the solution. Indeed, in the ideal context in which we know exactly all connections between the groups and the observed, we want to impose that precise structure on the network and therefore be strict on the regularisation.
The minimisation of Problem (41) uses the same minimisation schema of Algorithm 5 with the only difference that TGL_κ needs a weighted version in the ℓ_1 penalty. This is easily done given the separability of the ℓ_1 norm.

5.4.1 *Synthetic data experiments*

We compare the performance of MTGL_G with LTGL and MTGL_κ with discrete kernel. We want to assess that the availability of prior knowledge can be used to obtain better performance on the estimation of the latent layer while retaining good performance on the observed network. We generated data using

	r=4			r=20		
score	LTGL	MTGL^L	MTGL$_G$	LTGL	MTGL^L	MTGL$_G$
P	0.91 ± 0.01	0.93 ± 0.01	$\mathbf{0.93 \pm 0.01}$	$\mathbf{0.92 \pm 0.02}$	0.70 ± 0.03	0.70 ± 0.02
R	0.61 ± 0.01	$\mathbf{0.71 \pm 0.01}$	0.71 ± 0.02	0.61 ± 0.02	$\mathbf{0.87 \pm 0.09}$	0.86 ± 0.01
F1	0.73 ± 0.00	$\mathbf{0.81 \pm 0.00}$	0.80 ± 0.01	0.73 ± 0.01	$\mathbf{0.78 \pm 0.01}$	0.77 ± 0.01
S	$\mathbf{0.99 \pm 0.00}$	0.99 ± 0.00	0.99 ± 0.00	$\mathbf{0.99 \pm 0.00}$	0.99 ± 0.00	0.99 ± 0.00
BA	0.81 ± 0.00	$\mathbf{0.85 \pm 0.01}$	0.85 ± 0.01	0.80 ± 0.01	$\mathbf{0.93 \pm 0.00}$	0.92 ± 0.00
MSE$_{obs}$	$\mathbf{0.00 \pm 0.00}$	0.00 ± 0.00	0.05 ± 0.00	0.01 ± 0.00	$\mathbf{0.00 \pm 0.00}$	0.13 ± 0.07
MSE$_{lat}$	0.00 ± 0.00	$\mathbf{0.00 \pm 0.00}$	0.04 ± 0.00	0.00 ± 0.00	$\mathbf{0.00 \pm 0.00}$	0.13 ± 0.07
MRE	61.72 ± 0.38	fixed to 4	fixed to 4	61.54 ± 0.39	fixed to 20	fixed to 20

TABLE 9. *Average performance across 10 repetitions in terms of Precision (P), Recall (R), F_1-score (F_1), Specificity (S), Balanced Accuracy (BA), Mean Squared Error on the observed part (MSE_{obs}), MSE on the latent part (MSE_{lat} and Mean Rank Error (MRE) for the comparison of the Missing Temporal Graphical Lasso with Group imposition (MTGL$_G$), the Missing Temporal Graphical Lasso for Latent variables (MTGL$_L$) and the Latent Temporal Graphical Lasso (LTGL) all with discrete kernel on the observed part of the network for two datasets with a fixed number of observed variables $|O| = 200$ and latent variables set respectively to $r = 4$ and $r = 20$. Note that when variance equal to 0.00 is due to rounding to the significant digits.*

a diffusion evolution schema (see Appendix C.3) with $|O| = 200$ observed, $|M| = \{4, 20\}$ latent variables and $T = 10$ times.
The hyper-parameters are selected on the learning set by a Bayesian optimisation procedure based on expected improvement strategy (Snoek, Larochelle and Adams, 2012) through a 3-fold cross-validation. The score used is the generalised log-likelihood. Since MTGL$_\kappa$ is non-convex once we have the hyper-parameters we re-fit the model on the test set 10 times and we take the mean of the scores for the comparison in order to study the stability and reliability of the results.

5.4.2 *Results*

Group imposition does not effect the accuracy of structure inference on the observed part of the network but highly improves the recovery of the latent network
Table 9 shows the performance measures that are obtained for LTGL, MTGL$_L$ and MTGL$_G$ on the observed sub-network while Table 10 shows the performance in structure recovery on the latent network for both MTGL$_L$ and MTGL$_G$. Note that MTGL$_G$ estimates the latent variables if provided with their number, therefore, for comparison purposes, we had to impose the same number also to MTGL$_L$. We set this number to r, the true number of latent variables as previous experiments showed how little, setting different values of r, affects the final result.
Note that, scores in structure recovery on the observed part are quite similar across al three methods. We also observe that MTGL$_G$ has a higher MSE_{lat}

	r=4		r=20	
score	MTGL^L	MTGL$_G$	MTGL^L	MTGL$_G$
P	0.80 ± 0.38	$\mathbf{0.89 \pm 0.00}$	0.99 ± 0.01	$\mathbf{0.94 \pm 0.03}$
R	0.30 ± 0.35	$\mathbf{0.86 \pm 0.00}$	0.13 ± 0.03	$\mathbf{0.65 \pm 0.26}$
F1	0.20 ± 0.06	$\mathbf{0.88 \pm 0.00}$	0.23 ± 0.04	$\mathbf{0.72 \pm 0.24}$
S	0.8 ± 0.4	$\mathbf{0.99 \pm 0.00}$	0.99 ± 0.01	$\mathbf{0.99 \pm 0.00}$
BA	0.55 ± 0.02	$\mathbf{0.93 \pm 0.00}$	0.57 ± 0.02	$\mathbf{0.83 \pm 0.13}$

TABLE 10. *Average performance across 10 repetitions in terms of Precision (P), Recall (R), F_1-score (F_1), Specificity (S), Balanced Accuracy (BA), Mean Squared Error on the observed part (MSE_{obs}), MSE on the latent part (MSE_{lat} and Mean Rank Error (MRE) for the comparison of the Missing Temporal Graphical Lasso with Group imposition (MTGL$_G$), the Missing Temporal Graphical Lasso for Latent variables (MTGL$_L$) with discrete kernel on the latent part of the network for two datasets with a fixed number of observed variables $|O| = 200$ and latent variables set respectively to $r = 4$ and $r = 20$. Note that when variance equal to 0.00 is due to rounding to the significant digits.*

error that may be due to the higher sparsity of model, possibly causing higher values of the retrieved edges. Nevertheless, MTGL$_G$ is able to obtain optimal results in terms of structure recovery on the latent part (Table 10). In fact, in both experiments ($r = 4$ and $r = 20$) the F1 score is higher than 0.7 and the specificity is close to 1. When $r = 20$ the increase in the number of latent variables induces a decrease in recall which still has an acceptable value.
Conversely, MTGL^L does not perform well in terms of recall on the latent variable structure estimation. Indeed, without using the prior knowledge on the links, MTGL^L tends to infer an identity matrix on the $K_t[M]$ block, keeping a full matrix in the block $K_t[OM]$. This explains why the precision is high while the recall is extremely low.

5.5 Summary

In this chapter we presented two possible methods that deal with latent or partial data in the context of Gaussian Graphical Models. We presented one non-convex method based on EM algorithm and one convex approach optimised through ADMM. We also analysed the case in which prior knowledge is available on the latent variables that are seen as groups of the observed ones. Our LTGL$_\kappa$ method has already been exploited in (Chang, Yao and Allen, 2019) for the inference of brain connectivities which proved the efficacy of our method to deal with complex real-world data.
In the future we plan to provide theoretical bounds in terms of number of samples per each time N_t, number of missing variables $|M|$ and observed $|O|$ in order to have guarantees of an accurate inference. Indeed, we argue that the theoretical bounds provided in literature for the stationary case may be less

strict for dynamic network given the employment of temporal dependency and consistency.

6

REGAIN

REGAIN is short for *REgularised GrAph INference* and it is a vast Python library that provides a straightforward implementation of recent advances of graphical models as well as utilities and plotting functions for their assessment and visual representation.
The main minimisation algorithms that we exploit are the alternating direction method of multipliers (ADMM) (Boyd et al., 2011) and the forward-backward splitting (FBS) (Combettes and Wajs, 2005) for convex functionals. We use the Expectation Maximisation (EM) algorithm (Dempster, Laird and Rubin, 1977) for non-convex optimisation problems that typically involve missing data.
We designed REGAIN in such a way to be fully compatible with the Scikit-learn library (Pedregosa et al., 2011) which is the most used machine learning library in the Python language. This compatibility allows us to use the model selection methods, scoring utilities and many other tools that Scikit-learn offers. REGAIN is heavily based on the popular low-level numerical libraries for linear algebra Numpy (Oliphant, 2006–) and SCIPY (Jones, Oliphant and Peterson, 2001–).

OUTLINE This chapter is organised as follows. In Section 6.1 we list all the methods implemented within the REGAIN package. In Section 6.2 we present the related packages that implement similar or the same algorithms, and, in Section 6.3, we compare some of our implementations. In Section 6.4 we show how the library can be installed and in Section 6.5 an example of usage of the library. We conclude in Section 6.6 with a brief recap and further analysis to perform on the library.

6.1 Implemented models

REGAIN provides a great variety of algorithms that assume Gaussian distribution, indeed it contains the implementation for the recent proposed time-varying graphical models in (Hallac et al., 2017a; Tomasi et al., 2018a,b), steady-state graphical models (Chandrasekaran, Parrilo and Willsky, 2010; Friedman, Hastie and Tibshirani, 2008; Ma, Xue and Zou, 2013) as well as a Bayesian sub-

module where we implemented a Bayesian graphical lasso (BGL) following the procedure of (Moghaddam et al., 2009) and Wishart process (WP) (Wilson and Ghahramani, 2011).
It includes methods for the estimation of GGMS with missing data as the Latent variable Graphical Lasso (LGL) (Chandrasekaran, Parrilo and Willsky, 2010; Ma, Xue and Zou, 2013), the missing Graphical Lasso (MissGL) (Städler and Bühlmann, 2012) and its version for the latent variables (Yuan, 2012). REGAIN contains also the implementation of MRFs with other distribution assumptions (Yang et al., 2015), in particular at the moment it contains the Ising model (Ravikumar, Wainwright and Lafferty, 2010) and the Poisson model (Allen and Liu, 2013). The library includes all the temporal models presented in this thesis, and the notebooks that test them.
We provide in Table 11 and Table 12 the summary of all the models presented in the REGAIN library with all the features that identify them. We highlighted in blue the models, addition and minimisation algorithms that are an original contribution of this thesis.
The package includes utilities for data generation on all the considered distributions as well as different evolution schemas (see Appendix C). It provides all the proximal operators for the temporal consistency functions Ψ (see Section 1.7) and popular norms computation. REGAIN includes the stability-based model selection method generalised to any number of penalties and of networks (see Chapter 3) as well as the generalised scores based on the likelihood of the model. Note that, for any grid search or random search procedure it is sufficient to nest our implementations in Scikit-learn pipelines.

6.2 Related Packages

REGaIN inherits the structure and basic functionalities from the scikit-learn package. Scikit-learn includes the graphical lasso estimation, which is minimised using a coordinate descent algorithm. In this implementation we exploit the ADMM to have a uniform implemented throughout all implemented methods. For the latent variable graphical lasso, our implementation follows the model as proposed by (Chandrasekaran, Parrilo and Willsky, 2010) and implemented via ADMM by (Ma, Xue and Zou, 2013). Ma, Xue and Zou, 2013 included a link to download their Matlab implementation[1], but no open-source library includes such code. For MissGL with the version with latent variables there is a R package called LVGLASSO which is implemented using the expectation-maximisation (EM) method, as proposed by (Yuan, 2012). The joint graphical lasso is implemented in an R package [2]. The time-varying graphical lasso (Hallac et al., 2017a) has already been implemented in Python using ADMM. However, the implementation relies on CVXOPT[3], which has shown to be less optimal for the computational constraints and scalability with

1 https://www.math.ucdavis.edu/~sqma/ADMM-LVGLasso
2 https://rdrr.io/cran/JGL/man/JGL.html
3 https://cvxopt.org/

Method	Time-varying	Missing Variables	Latent variables	Kernel version	Minimisation procedure	References
GL					ADMM	(Friedman, Hastie and Tibshirani, 2008)
BGL					CD[a]	(Friedman, Hastie and Tibshirani, 2008)
JGL				✓	ADMM	(Danaher, Wang and Witten, 2014)
Miss-GL		✓	✓		EM + ADMM	(Städler and Bühlmann, 2012)
LGL			✓		ADMM	(Ma, Xue and Zou, 2013)
TGL	✓			✓	ADMM, FBS	(Hallac et al., 2017a; Tomasi et al., 2018a)
LTGL	✓		✓	✓	ADMM	(Tomasi et al., 2018b)
MTGL	✓	✓	✓	✓	EM + ADMM	(Städler and Bühlmann, 2012)
WP	✓			✓	MCMC	(Wilson and Ghahramani, 2011)

TABLE 11. *Summary of the available Gaussian Graphical Models in REGAIN.*

Method	Time-varying	Kernel version	Minimisation procedure	References
IGM			CD, FBS	(Ravikumar, Wainwright and Lafferty, 2010)
PGM			CD, FBS	(Allen and Liu, 2013)
TIGM	✓	✓	ADM + FBS	
TPGM	✓	✓	ADMM + FBS	

TABLE 12. *Summary of the available Generalised Graphical Models in REGAIN.*

a Coordinate descent, based on the graphical lasso as implemented in scikit-learn.

respect to plain Numpy and Scipy (Tomasi et al., 2018b). For the graphical models with other distribution based on generalised linear models there is a package XMRF in R (Wan et al., 2016) that provides both data generation and inference algorithms.

6.3 Scalability

We now want to provide an assessment of how the models perform in comparison with others in terms of time to convergence as the number of unknowns increases. All the compared methods are initialised in the same manner, *i.e.*, with all variable interactions set to zero. For all methods we fixed the maximum number of iterations to 100. We ran all experiments on a machine provided with two CPUs (2.4 GHz, 8 cores each).

COMPARISON OF LTGL, TGL AND MTGL. We performed a scalability analysis of the Latent Temporal Graphical Lasso (LTGL), our implementation of the TIme-varying Graphical Lasso (TGL) (Hallac et al., 2017a) and the Missing Temporal Graphical Lasso in the case of Latent (MTGL^L) and Partial (MTGL^P) variables. The first two are based on ADMM while the last is based on EM whose maximisation step consist in minimising a TGL functional. We generated data according to an ℓ_1 evolution schema (see Appendix C.1) with $T = 10$, $N_t = 100$ and $D = \{5, 10, 80, 200, 500, 1000\}$. The temporal complexity of LTGL and TGL are similar up to a constant while MTGL has higher complexity, the unknowns are $T\frac{D(D+1)}{2}$ for TGL, $2T\frac{D(D+1)}{2}$ for LTGL and $2\big(\frac{D(D+1)}{2} + D\big)$ for MTGL. Results are presented in Figure 33 and are in line to what we expected. Indeed, TGL has the lowest number of unknowns and the lowest time to converge that is quite similar to the LTGL one. MTGL^L and MTGL^P have the same trend ad the number of variable increases but have convergence time grater than 1 or 2 order of magnitude more than TGL. This is expected as well as the fact that MTGL^P has the worst performance considering it needs to also estimate the means. In Figure 34 we also plotted the number of iterations needed for convergence. Note that LTGL is the one requiring more iterations wile still having good time of convergence. MTGL^P, instead, requires few iterations but it contains a further nested minimisation of the TGL problem thus it requires more time to converge. MTGL^L is the most un-stable as it reaches the fixed maximum number ot iterations two times.

COMPARISON OF LTGL, LVGLASSO AND TGL. We performed a scalability analysis using Latent Temporal Graphical Lasso (LTGL) with respect to different ADMM-based solvers. We evaluated the performance of our method in relation to Latent Graphical Lasso (LVGLASSO) (Ma, Xue and Zou, 2013) and the Time-Varying Graphical Lasso (TGL) (Hallac et al., 2017a), both implemented with closed-form solutions to ADMM sub-problems. In general, the complexity of the three compared solvers is the same (up to a constant). We generated different data sets $X \in (\mathbb{R}^{N\times D})^T$ with different values of T and D. In

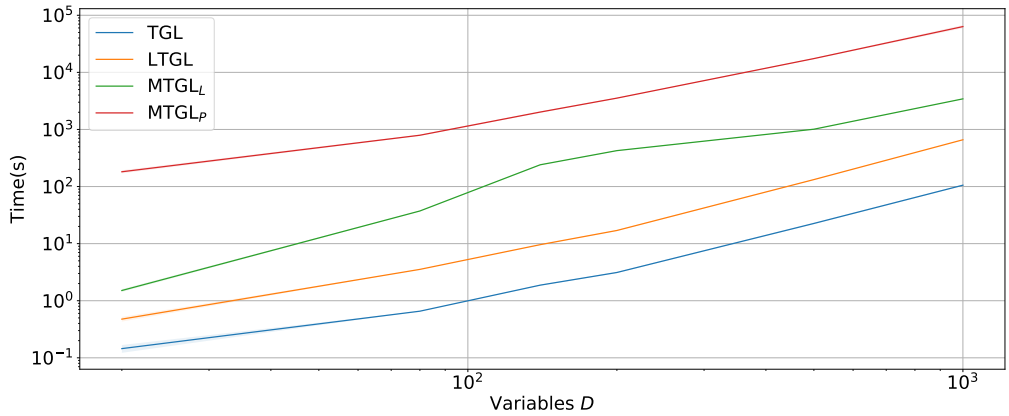


FIGURE 33. *Scalability comparison in terms of seconds for convergence of the Latent Temporal Graphical Lasso (LTGL), our implementation of the TIme-varying Graphical Lasso (TGL) and the Missing Temporal Graphical Lasso in the case of Latent (MTGL^L) and Partial (MTGL^P) variables.*

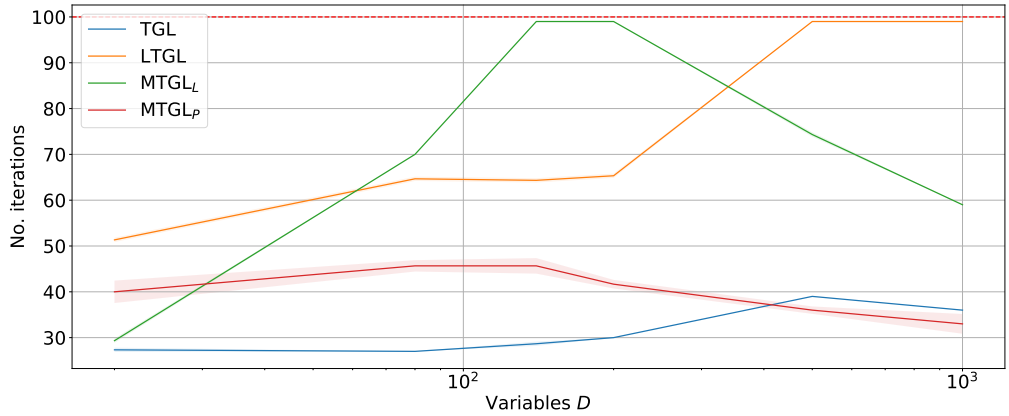


FIGURE 34. *Scalability comparison in terms of iterations for convergence of the Latent Temporal Graphical Lasso (LTGL), our implementation of the TIme-varying Graphical Lasso (TGL) and the Missing Temporal Graphical Lasso in the case of Latent (MTGL^L) and Partial (MTGL^P) variables.*

particular, $D \in [10, 400)$ and $T = \{20, 50, 100\}$. We ignored the computational time required for hyper-parameters selection.
Figure 35 shows, for the three different time settings, the scalability of the methods in terms of seconds per convergence considering different number of unknowns of the problem (*i.e.*, $2T\frac{D(D+1)}{2}$ with D observed variables and T times). In all settings, LTGL outperforms LVGLASSO and TVGL in terms of seconds per convergence. In particular, the computational time for convergence remains stable disregarding the number of time points under consideration.

COMPARISON OF TWO IMPLEMENTATIONS OF LGL We compared our implementation of the Latent Graphical Lasso (LGL) (Chandrasekaran, Parrilo and Willsky, 2010) with the one available online (LVGLASSO) (Yuan, 2012).

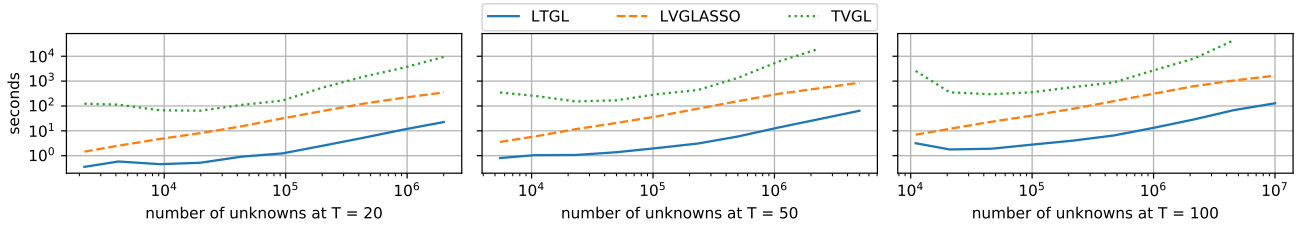


FIGURE 35. *Scalability comparison in terms of seconds for convergence of the Latent Temporal Graphical Lasso (LTGL), our implementation of the TIme-varying Graphical Lasso (TVGL) and the Latent Variable Graphical Lasso (LVGLASSO).*

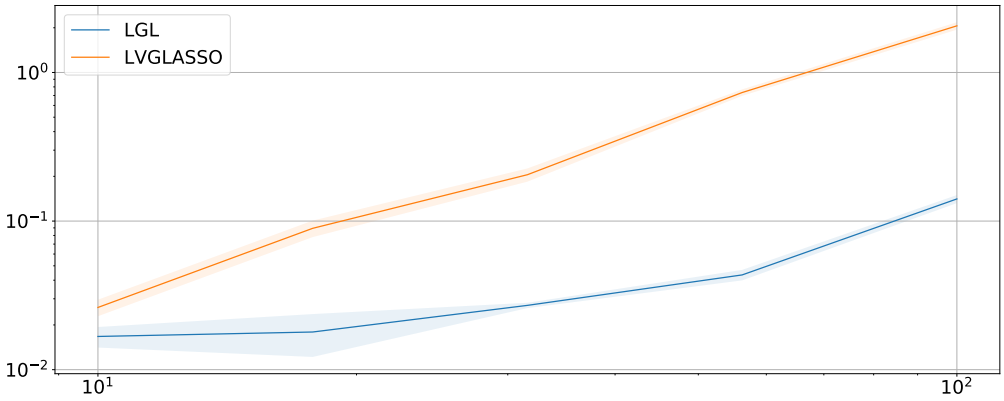


FIGURE 36. *Scalability comparison in terms of seconds for convergence of our implementation of the Latent Graphical Lasso (LGL) and the original (LVGLASSO).*

The first one is based on ADMM, the second one on EM. We generated stationary data for D that varies in the interval $[10, 100]$ and $N_t = 100$. Results are shown in Figure 36 where we can observe that the slope of LGL is less than the one of LVGLASSO. Also, LVGLASSO, while having a convergence time similar for $D = 10$, became rapidly distant with one order of magnitude difference.

COMPARISON OF TGL$_\kappa$, TIGM$_\kappa$, TPGM$_\kappa$. We analysed the time of convergence of the three TGL$_\kappa$, TIGM$_\kappa$, TPGM$_\kappa$ methods assuming different distributions. We generated the same data with $T = 10$, $N_t = 100$ and $D = \{5, 10, 80, 200, 500\}$. We then applied our implementations assuming an RBF kernel.
Results are in Figure 37 where we can see that while the ascendant trend is similar across all algorithms. TPGM$_\kappa$ is the one which requires more time to converge while TIGM$_\kappa$ and TGL$_\kappa$ show comparable performance.The worst performance of TIGM$_\kappa$ and TPGM$_\kappa$ is probably due to the non-closed form of one step of the ADMM minimisation procedure used for the optimisation that, thus, requires a nested minimisation algorithm that slows down convergence. This may improved by parallelising the algorithm.

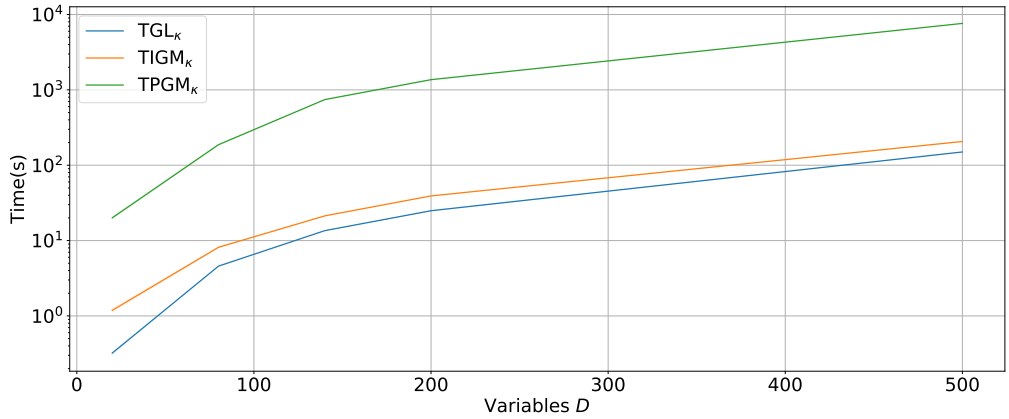


FIGURE 37. *Scalability comparison in terms of seconds for convergence of the Kernel Temporal Graphical Lasso (TGL$_\kappa$), the Kernel Temporal Ising Graphical Model (TIGM$_\kappa$) and the Kernel Temporal Poisson Graphical Model (TPGM$_\kappa$*

6.4 Installation

REGaIN is available as an open-source Python library, distributed under BSD-3-Clause, at `https://github.com/veronicatozzo/regain` or `https://github.com/fdtomasi/regain`. The library depends on Numpy, Scipy and scikit-learn. It can be installed via the Python package managers `pip` or `conda`:

```
$ pip install regain
```

or

```
$ conda install -c fdtomasi regain
```

Alternatively, REGaIN can be installed from source, using the following commands:

```
$ git clone https://github.com/veronicatozzo/regain
$ cd regain
$ python setup.py install
```

The library includes numerous Jupyter notebooks showing usage examples of the implemented classes.

6.5 Usage Example

The following example show a basic example of the usage of REGaIN, generating the data and inferring the precision matrix associated to the data under the influence of latent factors (see Chapter 5).

```
>>> import numpy as np
>>> from regain.covariance import LatentTimeGraphicalLasso
>>> from regain.datasets import make_dataset
>>> from regain.utils import error_norm_time
```

```
>>>
>>> np.random.seed(42)
>>> data = make_dataset(n_dim_lat=1, n_dim_obs=10)
>>> X = data.data
>>> theta = data.thetas
>>>
>>> mdl = LatentTimeGraphicalLasso(max_iter=50).fit(X)
>>> print("Error: %.2f" % error_norm_time(theta, mdl.precision_))
```

6.6 Summary

The presented library contains many algorithms, some original of REGAIN and others present only in R or Matlab. We aim at improving the optimisation of some of the implemented algorithm by exploiting the separability of the variables by parallelising the algorithm, and, thus, accelerating their time to converge. We argue that REGAIN provides a useful tools to researchers in the graphical modelling field who wants to use Python, one of the most used language for machine learning.
We are still working on the library adding features on a daily basis. We plan on further validating the presented methods to find eventual bugs and we plan to reach a stable release in the next few months.

PART III

Applications and conclusions

Part 3 includes some applications and the conclusions of this thesis. Chapter 7 described applications of the previously introduces methods on real-world datasets. Chapter 8 presents a recap of the proposed work with some lines for future research directions.

7

Real-world applications

During the development of the thesis we applied our inference methods on some real-world dataset to assess their ability to detect patterns or infer interesting connections between variables. In particular we used two datasets on which we applied more than one method:

1. food search trends that we analysed with MTGL_G and TGL_P;
2. stock market prices that we analysed with MTGL_G and LTGL.

Then, we also applied $MTGL_L$ with prior on Neuroblastoma gene expression profiles, and, $LTGL_\kappa$ on weather data. We would like to point out that, while the majority of methods are designed with biological applications in mind, is not quite easy to obtain public temporal dataset of this type. Therefore we had to rely on other type of data.
Nonetheless, on all applications we retrieved interpretable patterns that reinforce our beliefs on the utility of our proposed methods for the analysis of real-world phenomena. We want to point out that the only application that had the support of an a posteriori analysis by an expert in the field is the one on Neuroblastoma data, while the other applications were more of support of the method.

OUTLINE The rest of this chapter is organised as follows. In Section 7.1 we show the analysis of food trend search. In Section 7.2 we present results obtained analysing stock market prices with different methods. In Section 7.3 we present results obtained with the groups imposition on a stationary neuroblastoma dataset. In Section 7.4 we present results obtained on weather measuring sensors data. We conclude in Section 7.5 with a summary of the chapter and future work directions.

7.1 Food search trends

We analysed Google© food trend search data. We downloaded the food dataset from `http://rhythm-of-food.net`. It contains records of food searches in the

US from 2004 to 2016, in particular data is formed by weekly scores of 201 different food. We discarded four as they were specific for regions outside the US, ending with 197 variables.
The aim of our analysis is to detect similarities within a year, therefore, given the high annual periodicity of the data we merged years together in order to have more samples per time point ending up with 52 time points on 197 observed variables with 13 samples per time point. We applied Log2 transform to ensure normality of the data. We analysed these data with two methods: MTGL_G in which we imposed group knowledge on the data in terms of food group *e.g.*, artichokes and aubergine are vegetables while chicken and pork are meat. Ideally, we aim at finding correlations between data that are explained by the food group rather than by the specific food. We used food groups downloaded from `foodb.ca` eliminating those groups that had no connections with the 197 food ending up with 18 groups.
Given the non-convexity of MTGL_G we repeated the experiments multiple times to get mean results. Then we analysed the same data with TGL_P trying to infer, only from observations, similarity patterns across the weeks of the year.
For both experiments we utilised a ℓ_2^2 temporal consistency function. In order to find the best hyper-parameters we used Bayesian search on all data.

RESULTS

Results show that people tend to search similar terms before any major holiday. They also show that the group prior imposition allows to obtain interesting insights that are in line with results obtained with TGL_P.

In Figure 38 we present the results obtained by applying MTGL_G. In particular we depicted the non-constantly zero correlations between groups of foods obtaining 15 interesting mean trends over 10 repetitions of the experiments. These curves show interpretable peaks. We note that major peaks are present right before the major US holidays. Let us consider the correlation meat-baking depicted in panel (e), here we can observe three distinct peaks right before Memorial Day (pink vertical line), Thanksgiving (black vertical line) and Christmas (orange vertical line) which may possibly indicate a correlation in research for typical baked holidays plates based on meat. Another interesting correlation can be found in panel (h) that corresponds to meat-beverages in the spring period. We argue that, in this period people tend to have more barbecue and therefore try best combinations of meat and beverages. We also found obvious results in panel (i) that shows a positive correlation, always non-zero, among dishes and herbs and spices. Clearly people tend to search for the best spices to put in recipes throughout the years and not only in specific periods. Another similar behaviour can be found in panel (n) fruit-vegetables.
The results of the application of TGL_P are depicted in Figure 39 where we represented a circular heat-map representing a hierarchical clustering of the weeks. Each layer considers an increased number of clusters. Hence, the outer layer has the finest-grain on the patterns in time. If we consider the outer layer we can observe that the weeks corresponding to holidays periods or fest-

ivity are clustered together (light green cluster), indicating that people tend to search traditional recipes terms in those periods. We refer for example to the weeks during Christmas (52-1), Thanksgiving (47-48), Easter (15-16) and many others. This result is in line with the results obtained with MTGL_G that showed peaks right before holidays periods meaning that there is a sort of similar behaviour in the search trends in those periods of the year.
We then observed in details two specific networks corresponding to the light green cluster (that we identified as the holiday cluster) and then we also observed the network corresponding to the blue cluster. We plot only the common edges among the networks in the cluster in Figure 40 and 41. Recall that, TGL_P does not force networks to be identical, therefore, some edges may be added to the plotted networks to retrieve the members of the cluster. In the first one we can observe that one major hubs related to the search of term *macaron* and a second one is related the term *kale*. While the first hub seems surprising giving the french nature of these cookies, we discovered that macarons gained a lot of attention in North America in the 2010s, becoming one of the most common sweet. The second hub is less surprising as some of the typical holidays recipes in US are based on kale. This is true also for *brussel sprouts*.
The hubs related to *moscow-mule*, *quinoa* and *chia* and also *kale* are present in the majority of the networks over the 52 weeks. Indeed, we can also observe them in Figure 41. For the last two terms we impute this fact to these last years tendency of healthier diets, that introduced these types of aliments in daily regimes. Regarding the Moscow mule cocktail, we argue it is due to the US origin of this cocktail that was born in the 1940s but regained popular interested in 2007.

7.2 Stock market prices

Finance is another example of a complex dynamical system suitable to be analysed through a graphical model. Stock prices, in particular, are highly related to each other and subject to time and environmental changes, *i.e.*, events that modify the system behaviour but are not directly related to companies share values (Bai and Ng, 2006). Here, the assumption is that each company, while being part of a global financial system, is directly dependent from only a subset of others. For example, it is reasonable to expect that stock prices of a technology company are not directly influenced by trend of companies on the primary sector. In order to show this, we analysed stock prices in a period ranging from 1998-2013. We analysed them with MTGL_G with ℓ_2^2 temporal consistency imposing, as a prior, the sector to which the companies belonged. We downloaded the sectors from `https://datahub.io/core/s-and-p-500-companies`. Then, on the ICT sector, we exploited LTGL to perform a further analysis during the financial crisis of 2007-2008. Data were downloaded from `https://quantquote.com/historical-stock-data`. We used

a a group lasso (ℓ_2) penalty to detect global shifts of the network. We used Bayesian search to identify the best hyper-parameters of the model.

RESULTS

Latent variables allow to detect triggering events of the 2008 financial crisis while prior on sectors break the latent variables influence in more interpretable facts.
Some of these results were partially presented in (Tomasi et al., 2018b).
Figure 42 shows the results obtained using MTGL$_G$, in particular the global behaviour of latent part of the system is shown in panel B. By looking at panel A we can see that such behaviour splits in groups that reveal interesting insights on the data.. In particular we can see that there is a high correlation in the late 90s between the Information Technology sector and Financials, Health-Care, Consumer Discretionary, Energy and Industrials that is in line with the spread of technology in the market that changed the products. This belief is reinforced by the stable correlation that we have with sectors like Materials or Real Estate that are not heavily influences by ICT. In all the relations, though, we observe abrupt changes in the period right after the crisis (highlighted in orange). This period changes the equilibrium of the majority of the companies in the world so it is difficult to say something particularly specific. Therefore, we further analyse such period with LTGL only for the companies belonging to the ICT sector. Results are shown in Figure 43 were we can observe two major changes in both components of the network (latent and observed), in correspondence of late 2007 and late 2008. In particular, during October 2008 a global crisis of the market occurred, and this effect is especially evident for the shift of latent variables. Also, the observed network changes in correspondence of the latent variables shift or immediately after, caused by the effect of the crisis on the stock market. The latent factors influence explains how the change of the network was due to external factors that globally affected the market, and not to normal evolution of companies relationships. We further investigated on the causes for the first shift. Indeed, we found that in late 2007 it happened a drop of a big American company that was later pointed out as the beginning of the global crisis of the following year.

7.3 Neuroblastoma gene expression profiles

Genetic data are the main data we have in mind when we developed all the methods presented in this thesis. Nevertheless, we could not find recent time-series dataset on these type of data. Thus, we show one analysis performed on a stationary one. We applied MTGL$_G$ on Neuroblastoma stationary RNA-Seq data downloaded from `https://portal.gdc.cancer.gov/projects/TARGET-NBL`.
For computational ease, we considered a subset of genes known in literature to be involved in Neuroblastoma disease based on Phenopedia (Yu et al., 2010). The resulting list of 203 genes was provided to Webgestalt (Wang et al., 2013) for a functional characterisation through a gene enrichment analysis. We ended

up considering 116 KEGG (Kyoto Encyclopedia of Genes and Genomes) (Kanehisa and Goto, 2000) pathways where the subset of genes was found enriched. We then applied MTGL_G to this data by imputing the empirical covariance matrix of the gene expression and the membership of each gene to one or more pathways. Given the non-convexity of our model we optimise it 20 times with different initialisations, which led to different solutions. We retained the links present at least 70% of the 20 times.

RESULTS

Retrieved pathway-pathway interactions are significant for Neuroblastoma as well as hub genes in the co-expression network.

These results were previously presented in (Tozzo et al., 2018).

Figure 44 shows the pathway-pathway interactions while Figure 45 shows the gene-gene interactions. The inferred co-expression network (Figure 45) includes four common genes that emerge above others. These genes are PLEKHA4, IL6, S100B and NTRK. While the relevance of IL6 (Totaro et al., 2013; Zhao et al., 2018) and NTRK (Lipska et al., 2009) in neuroblastoma is a known fact, the role of the remaining two genes is still under investigation. Differently from PLEKHA4 which is poorly annotated, S100B is a well characterized gene whose chromosomal rearrangements and altered expression are known to be implicated in several neurological, neoplastic, and other types of diseases, including Alzheimer's disease, Down's syndrome and epilepsy. The hubs genes are: MICA, EGFR, NEFL, MYO10, VDR, HFE, HLA-C, GHR, PDLIM1, APOE, DAB2, PALU, CEBPA, IL-33. The involvement in Neuroblastoma of the first 7 genes of this list is present in literature (Bini et al., 2012; Borriello et al., 2016; Capasso et al., 2014; Cheng et al., 1996; Mrowczynski et al., 2017; Wang et al., 2018). Also Figure 44 shows that "Alzheimer's disease" and "One carbon pool by folate" are the two most strongly connected pathways. Folic acid has been recently connected to childhood cancer (Moulik, Kumar and Agrawal, 2017), while the one-carbon pathway was linked to Alzheimer's (Fuso et al., 2011). We also have a clique between "Wnt signaling", "Ubiquitin mediated proteolysis" and "One carbon pool by folate" pathway. It is known that WNT signaling pathway plays significant roles in the survival, proliferation, and differentiation of human neuroblastoma (Suebsoonthron et al., 2017) and "Ubiquitin mediated proteolysis" is crucial in the regulated degradation of proteins involved in neuroblastoma proliferation and survival (Hämmerle et al., 2013).

7.4 Weather data

We applied LTGL_{ESS} to sensor readings measuring every 5 minutes average temperature, light and humidity during summer in Melbourne (Australia). We considered readings in the location of Docklands Library, from December 15th, 2014 to January 17th, 2015 to avoid missing values, and we grouped together time stamps of available data for each hour resulting in 1-hour-long granu-

larity. Such data exhibit intrinsic recurrent temporal patterns over the days. However, the patterns cannot be detected by only considering a two hour-long sliding window (consecutive time points), while a periodic kernel spanning 24 time points would possibly lead to a reliable inference of the underlying system. We used LTGL_{ESS} with a periodicity of 24 hours to be able to detect 1-day long patterns and remove external influence on the time series (*e.g.*, the particular month of the year).

RESULTS

High periodicity in the data can be exploited to retrieve recurring behaviour in the data. Sensors weather data reveal a change in the temperature-light correlation every day at 6AM and 18PM. In turn, temperature and humidity change their correlation at 12PM and 6AM.

Figure 46 shows the results on the sensor readings data, in particular the estimated covariance, precision and latent factor contribution matrices. Note that the contribution of the latent variables in the covariance matrices is not factorised out. In this case the covariance is given by the inverse of the difference between the precision and latent contribution matrices at each time point. As expected, temperature and light have always a positive correlation over time. Instead, the humidity have an inverse correlation both with light and temperature. While the covariance includes the correlation between all pairs of variables, the precision matrix includes the information on the graph between such variables, that is their conditional independence. Indeed, plotting the precision matrices over time (Figure 46, middle row) shows Light to be conditionally independent from Humidity given the information coming from Temperature, as its value is almost null over time. Conversely, Temperature is conditionally dependent on Light. Also in this case, the results are in line with the intuition that the effect of Light on Humidity is not as strong as the effect of Temperature. The latent factors contribution over time (Figure 46, bottom row) shows a both positive and negative effect on Temperature-Humidity: positive during the day, negative during the night.

Figure 47 in a shorter period of time that allows us to see more details. As by previous intuition, temperature and light have always a positive correlation over time (note that negative values in the precision matrix correspond to positive correlation). Also, light is shown to be conditionally independent from humidity given the information coming from temperature, as the correspondent entries are almost null over time. Again, the results are in line with the intuition that the effect of light on humidity is not as strong as the effect of temperature, and therefore values of humidity are explained in terms of temperature and not on light. Moreover, considering a periodic kernel over time we note that the variables change their relations at 5am (between temperature and humidity) and at 8am and 5pm (between temperature and light), which are the inflection points in which the atmospheric conditions change every day. We observed that using our kernel-based method that incorporates periodicity is beneficial to understand external variables affecting the system, such as, in

this case, the particular time of the day in which the system is sampled, that otherwise could not be captured by only considering subsequent time points.

7.5 Summary

In this chapter we showed some real-world data analysis example where we mainly applied Gaussian-based models. Given the generality of our proposed models there are many datasets suitable to be analysed with our methods. We plan to exploit the temporal Ising model to analyse neural data as well as the temporal Poisson model for genomic data. Nonetheless, as previously mentioned, it is generally difficult to find open temporal dataset of molecular biology problems that are the most suited for TPGM_K.

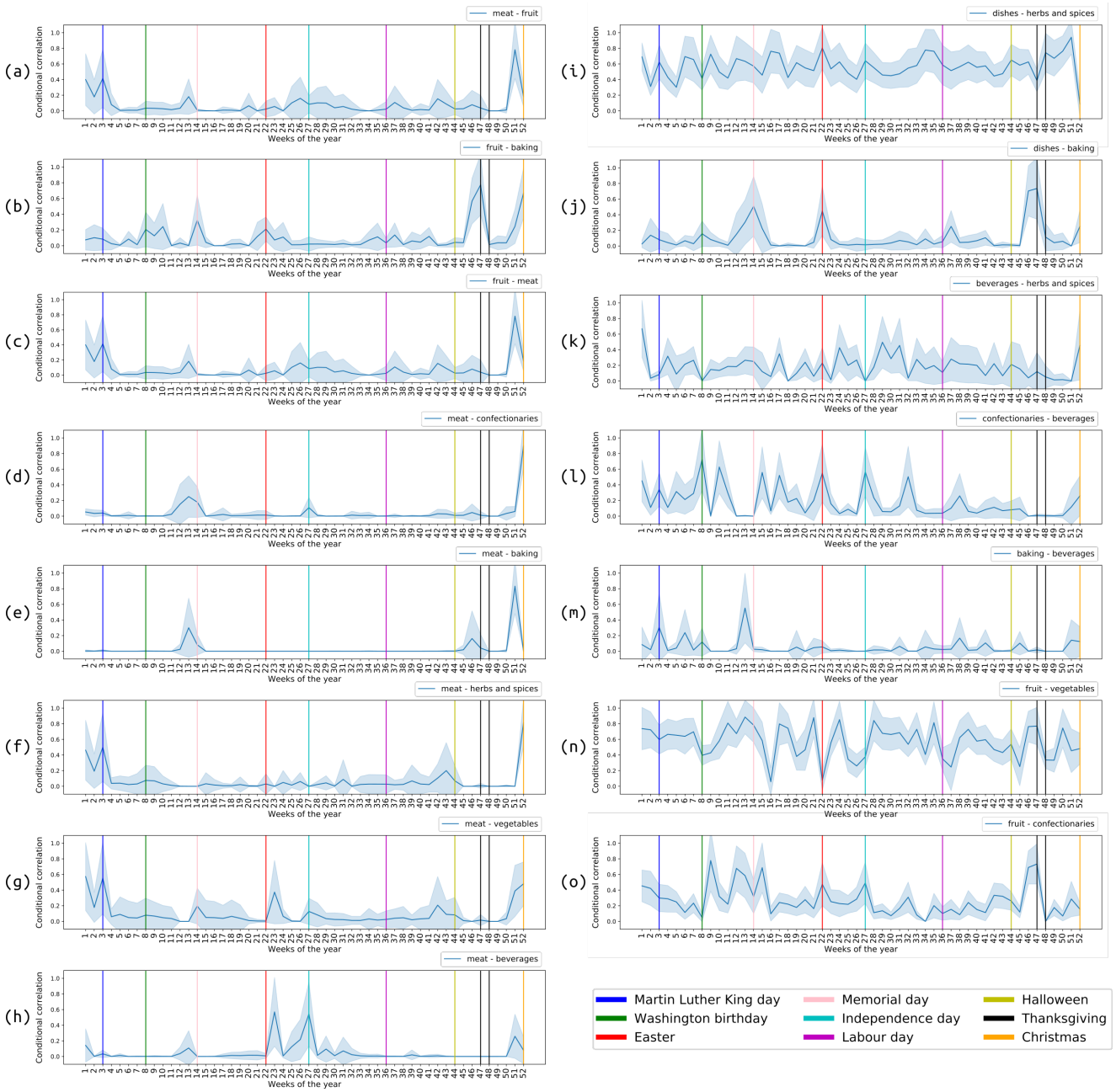


FIGURE 38. *Temporal correlations between 15pairs of food groups that showed a non-constant zero correlation in time. We repeated the analysis 10 times and show mean and standard deviation of the temporal behaviour. With the vertical coloured lines we indicate the periods of major holidays in US, as we noticed that for some relations there are interesting peaks right before these periods.*

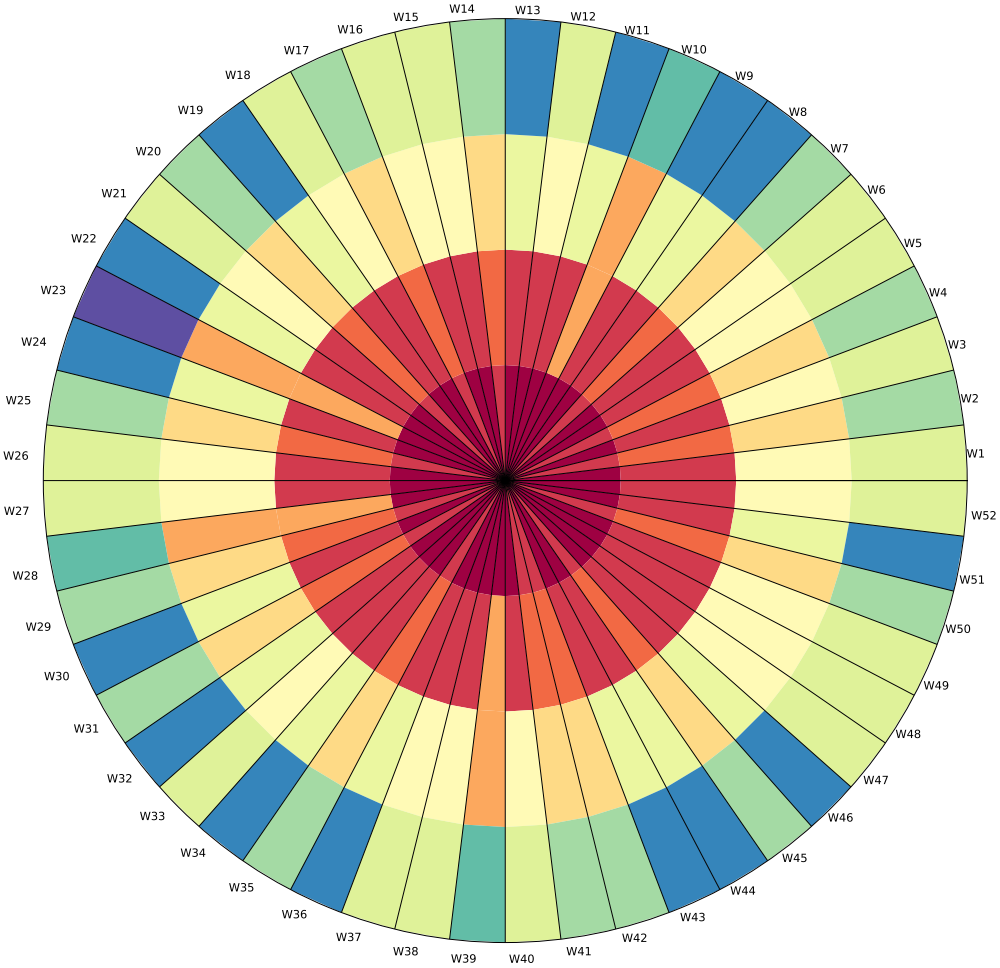


FIGURE 39. *Hierarchical clustering representation of weeks of the year obtained analysing inferred adjacency matrices on food search trends. Each layer corresponds to a different number of clusters, that increases going from the centre to outside letting explore clustering behaviour at various scales.*

m

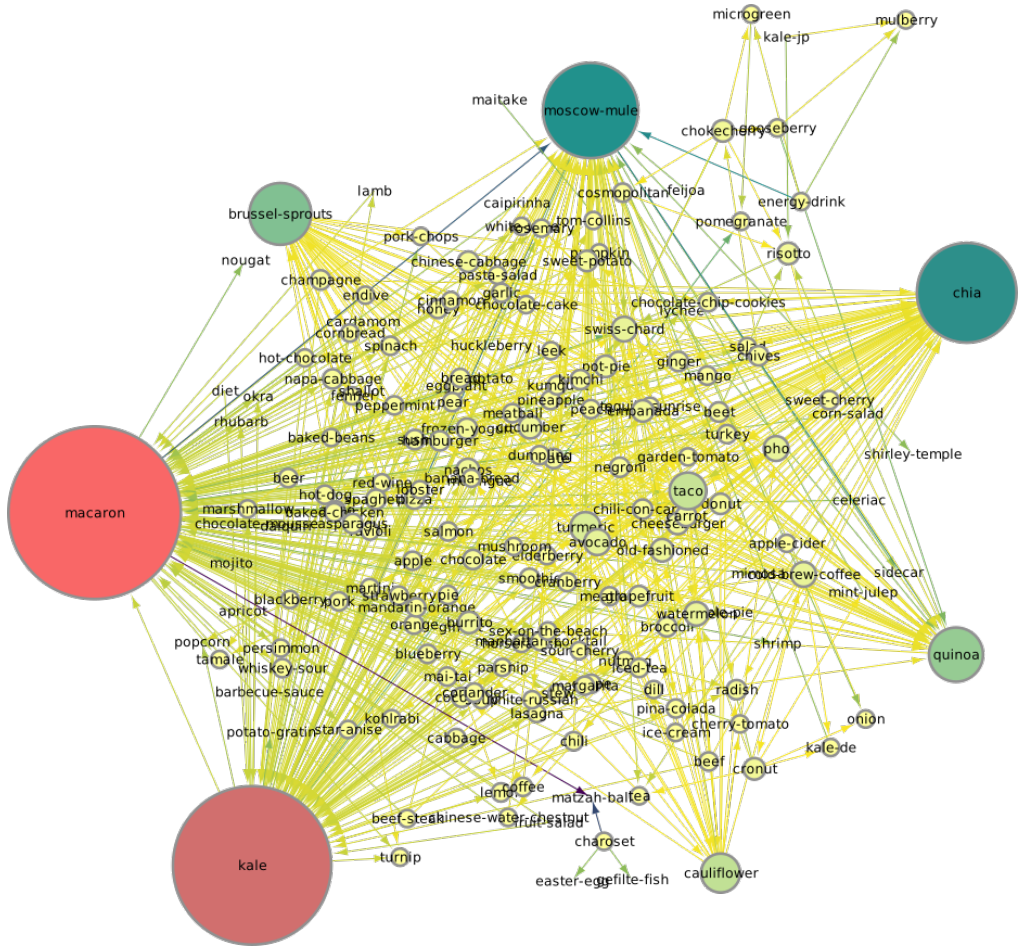


FIGURE 40. *Inferred network related to holidays weeks (light green cluster of Figure 39) considering nodes degree higher than 4 and their connected 1-degree neighbours. The hubs, in order of degree, are the following terms: macaron, cauliflower, moscow-mule, quinoa, taco, brussel sprouts, kale, chia.*

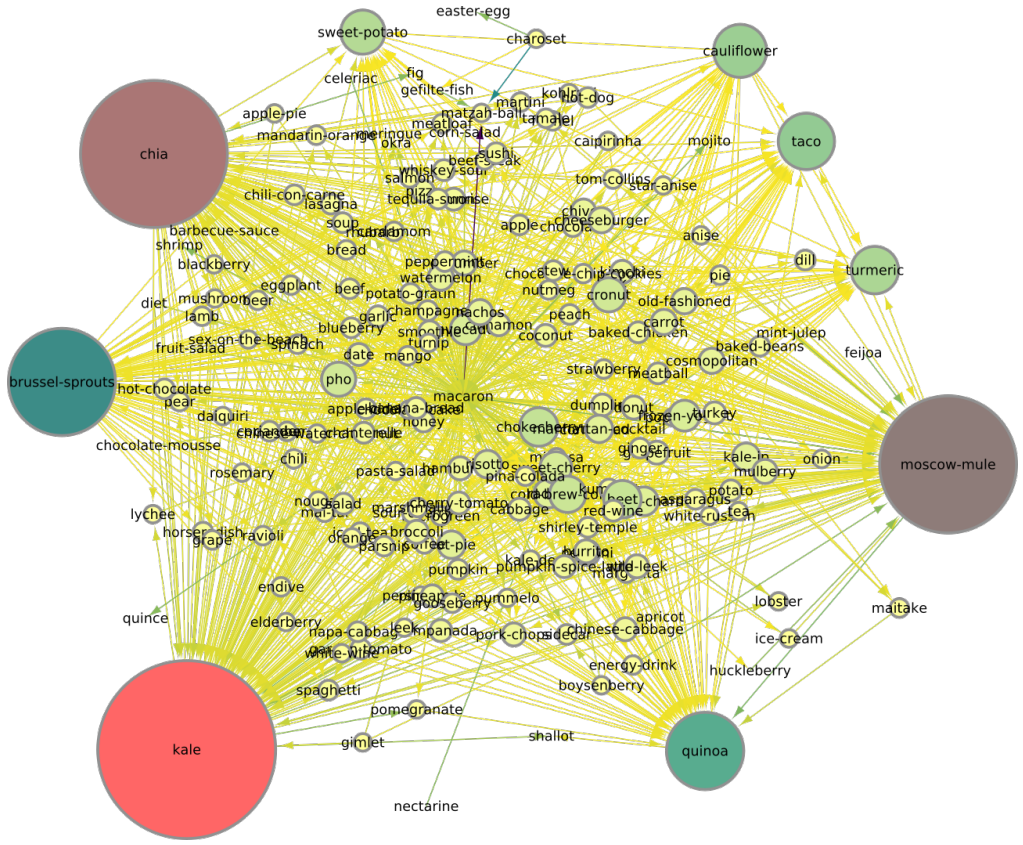


FIGURE 41. *Inferred network related to holidays weeks (blue cluster of Figure 39) considering nodes degree higher than 4 and their connected 1-degree neighbours. The hubs, in order of degree, are the following terms: kale, chia, moscow-mule, brussel sprouts, quinoa.*

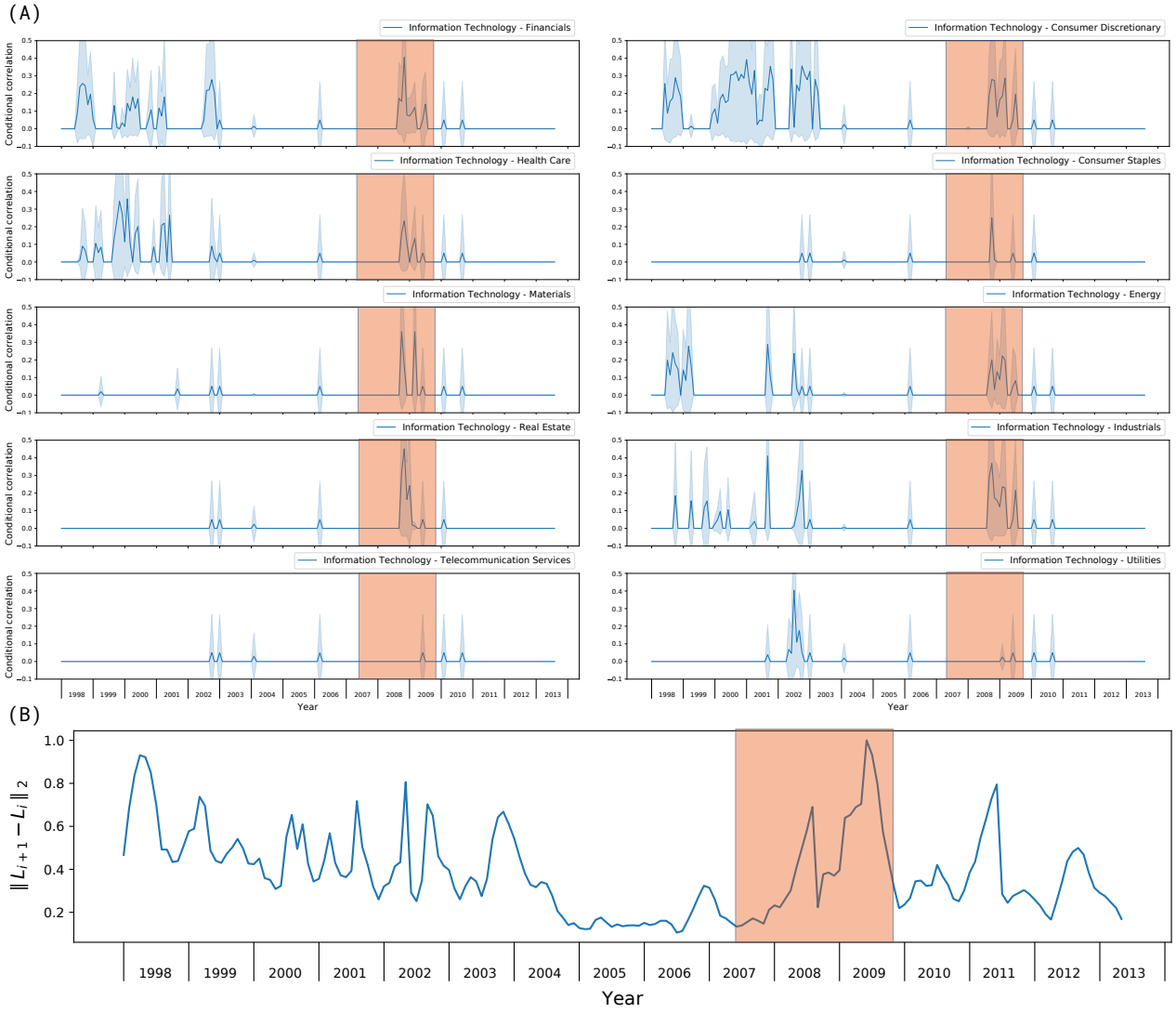


FIGURE 42. *Results obtained applying the Missing Temporal Graphical Lasso with Group imposition on stock market prices in the period 1998-2013 with group prior on their industrial sectors. In panel (A) we show the temporal conditional correlations between the ICT sector and the others, while in panel (B) we show the global temporal deviation of the latent marginalisation on all sectors. In orange we highlighted the period of the financial crisis.*

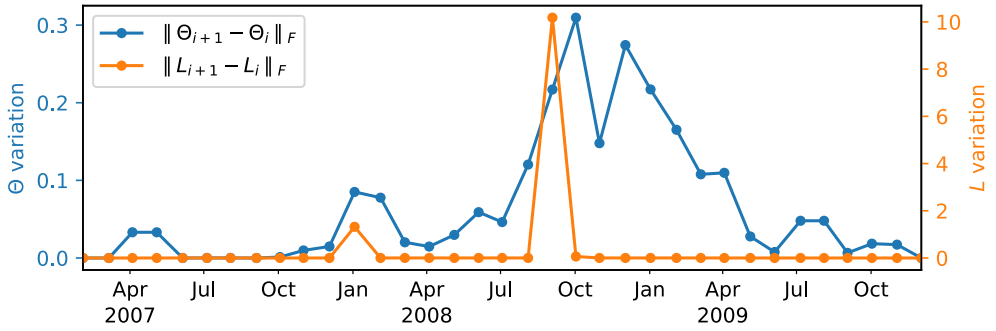
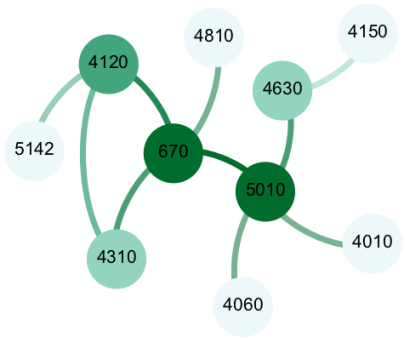


FIGURE 43. *Temporal deviation for stock market data in the period of time 2007-2009. Two peaks are present in correspondence of late 2007 and late 2008, in particular they correspond to the subprime mortgage crisis and the later Lehman Brothers collapse that are the two major trigger events of the financial crisis.*



No.	Description
4060	Cytokine cytokine receptor interaction
5010	Alzheimer's disease
670	One carbon pool by folate
4630	jak STAT signaling pathway
4120	Ubiquitin mediated proteolysis
5142	Chagas disease american trypanosomiasis
4010	MAPK signaling pathway
4150	mTor signaling pathway
4310	wnt signaling pathway
4810	Regulation of actin cytoskeleton

FIGURE 44. *Pathway-pathway interaction network obtained analysing Neuroblastoma TCGA data with Missing Temporal Graphical Lasso with Group imposition with KEGG pathways as group prior. The darker colour of the node denotes its degree while the darker colour of the edge denotes the probability of its existence.*

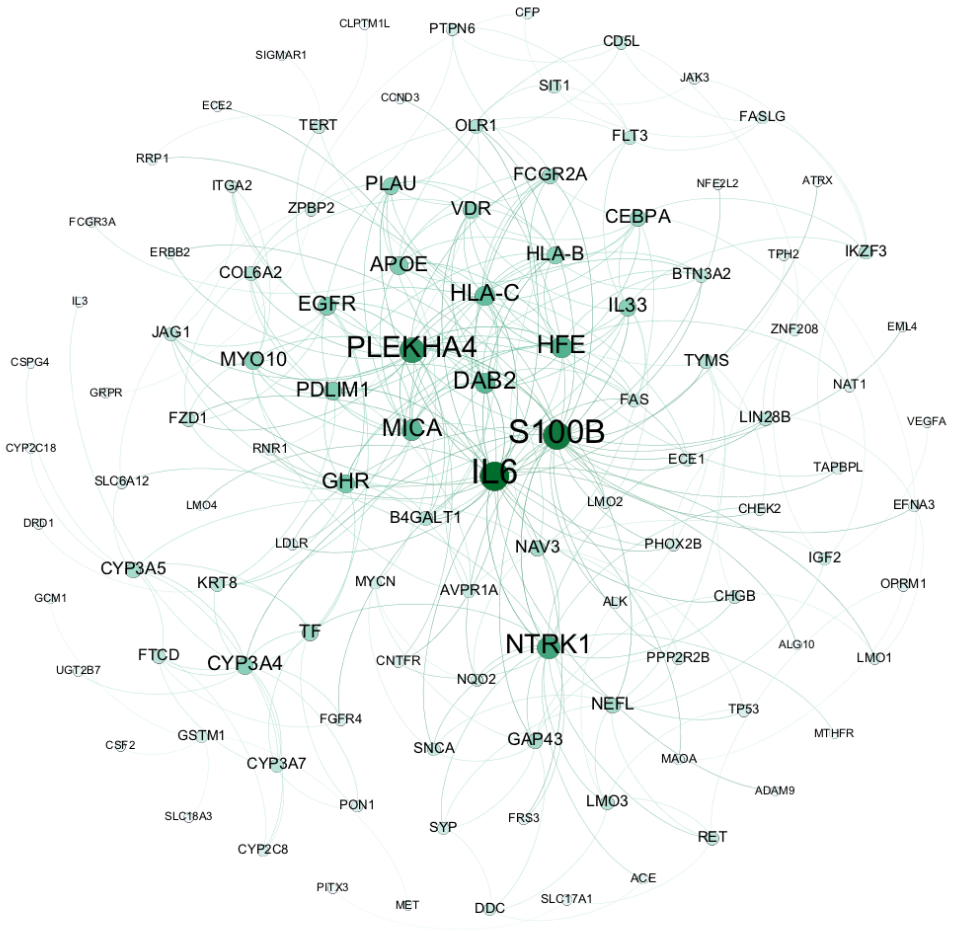


FIGURE 45. *Gene-gene interaction network obtained analysing Neuroblastoma TCGA data with Missing Temporal Graphical Lasso with Group imposition. The darker colour as well as the dimension of a node denote its degree while the darker colour of the edge denotes the probability of its existence.*

FIGURE 46. *Covariance, precision and latent marginalisation temporal values obtained applying Latent Temporal Graphical Lasso with periodic kernel ($LTGL_{ESS}$) on 15 days span of sensor for humidity, light and temperature in a location in Melbourne.*

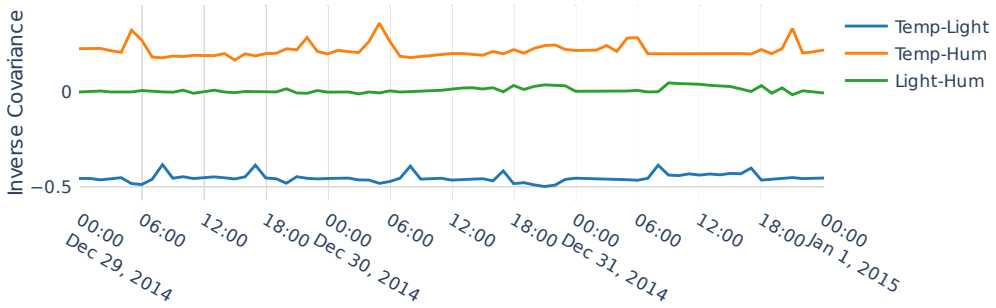


FIGURE 47. *Precision temporal values obtained applying Latent Temporal Graphical Lasso with periodic kernel ($LTGL_{ESS}$) on 4 days span of sensor for humidity, light and temperature in a location in Melbourne.*

Conclusions

We conclude this thesis presenting a brief recap of the major contributions as well as future research directions.
This thesis focuses on general network inference methods able to modelling temporal observations of real-world phenomena. In the last years there has been a major increase in the availability of multi-variate time-series data sets that possibly contain diverse data types (*e.g.*, neural spikes - binary, genomic - counts, sensors - continuous). Such data may have complex and intertwined temporal dependencies that may or not be known *a priori*. To the aim of modelling such data, we propose a general model for the inference of dynamical graphical models from multi-variate time-series (Chapter 4). Such model is suitable for different probability assumptions, temporal consistency function and temporal dependency patterns. We proposed a related minimisation algorithm that can be easily adapted to more distributions belonging to the exponential family class.
The second main focus of the thesis are missing data in the form of partial and latent variables. To cope with this type of data, that are naturally present in real-world measurements, we propose a temporal network inference method in the specific case of dynamical GGMs as the Gaussian assumption allows to easily deal with the problem. We are aware, though, of the need of dealing with this type of problem also for other distributions. Chen et al., 2016; Nussbaum and Giesen, 2019 for the Ising model, and, Vinci et al., 2018 for the Poisson model, proposed a network inference method with Gaussian latent variables solved via the same marginalisation proposed in Chandrasekaran, Parrilo and Willsky, 2010. We exploit such marginalisation for the development of the our LTGL model in Section 5.3, therefore, a possible future work could the extension to latent variables also for other distributions.
We validated the proposed methods on synthetic and, partially, on real data. While we plan to extend their experimental validation we also want to study their theoretical properties deriving from the assumption of non-stationary distributions connected trough a consistency function (Tran and Jung, 2018). We expect the bounds for sparsistency (*i.e.*, the sparsity pattern of the graph) to be less restrictive than the ones in the stationary case (Ravikumar et al., 2009a; Ravikumar, Wainwright and Lafferty, 2010; Yang et al., 2013, 2015). This is due to the fact that temporal dependency and consistency leverage on knowledge from dependent time points thus allowing for the use of less samples to retrieve the network structure with the same accuracy. While we have empirical proof of such phenomenon we want to analyse it more rigorously.
We also plan to develop an on-line inference method that, given the previous states of a network, is able to predict the network at time $t+1$, possibly guided by some measurements at that time (Chen, Meng and Zhang, 2019). We aim

at applying this prediction together with a regression technique for complex regression tasks. In particular we want to estimate the daily cost of energy that is needed to industries to propose a market price to the energy they sell.
From a more applied perspective we want to employ our temporal Ising model on neural data. The validity of using graphical models for the inference of neural functional connections has been extensively proved (Belilovsky, Varoquaux and Blaschko, 2016; Chang, Yao and Allen, 2019) and in particular the ability of the Ising model to correctly represent neural connections (Schneidman et al., 2006). We want to assess the validity of our method on neural data from patients subject to different stimuli at known time points (Somatosensory evoked data). Then, if the method with automatic kernel inferenc, correctly identifies the different stimuli, we plan on using such method on focal epilepsy data to detect interesting functional connections that may help in the identification of onset zone (Marsh et al., 2010).

We want to remark that research related to graphical modelling of time-series with and without missing data is still in its early days. The methods proposed in this thesis are an attempt to study this topic under different perspectives. Although not conclusive, this thesis points out new challenges and possible new possible solutions for the analysis of the increasingly available time-series datasets. In particular, there are three main aspects that are strictly related to the thesis and still open for further research: algorithmic complexity, time division, and causality.
Further improvements could be done regarding complexity to reach solutions in less time or iterations. We tried different approaches in Appendix A for specific settings of convex Gaussian models. Nevertheless, other methods could be employed for the optimisation of non-convex models. In particular, the EM algorithm could be optimised via typically faster Variational Inference methods (Bernardo et al., 2003; Blei, Kucukelbir and McAuliffe, 2017).
The problem related to time division lies in the main assumption of the proposed methods: time can be divided a priori in chunks without introducing bias. This is a strong assumption and should be further investigated and explored, possibly by inferring a network at each time point rather than at each chunk. Moreover, it would be interesting to extend the inference to an on-line version that allows for prediction of future time points.
Finally, a concept that is strongly related to this thesis is *causality* as it follows naturally from the modelling of the same variables in time. Nonetheless, the relation between Markov Random Models and causal graph is non-trivial. Causal graphs, typically called Bayesian Networks (Murphy and Russell, 2002), are defined as directed acyclic graphs that satisfy the Markov property. Thus, the two probabilistic models have a discrepancy in the structure definition which results in MRFs modelling correlation and Bayesian Networks modelling causation (Elwert, 2013). We suppose that a step towards causality may be done by conditioning each time stamp with respect to the previous ones, thus removing the edges possibly caused by some temporal effect, but this idea needs to be further refined.

To conclude, we argue that the methods developed in this thesis are a valuable contribute that lies in a wider set of models appropriate for time series and data analysis, which aim at a better understanding of underlying complex processes.

PART IV

Appendix

A

Minimisation of TGGM$_\kappa$, TIGM$_\kappa$, TPGM$_\kappa$

Problems TGL$_\kappa$, TIGM$_\kappa$, TPGM$_\kappa$ respectively Equations (34), (35) and (36) are convex, providing that the penalty function Ψ is convex, and coercive because of the regularisers, hence retaining convergence guarantees to the global optima.. In practice, however, finding such solution may be not trivial due to the high number of unknowns of the problem. Considering T time points and D variables, the total number of unknowns is $TD(D+1)/2$. We resort to the Alternating Direction Method of Multipliers (ADMM) (Boyd et al., 2011), that has been shown to be suitable to minimise complex functionals subject to constraints.

The ADMM procedure divides the original problem in sub-problems easier to minimise. Thanks to this separation the minimisation of the three different functionals in Equations (34), (35) and (36) is equal except for the step containing the likelihood. It also allows for more flexibility in the temporal patterns that the variable interactions follow.

Let us define two projections:

$$\begin{aligned} P_{Lm}\colon (\mathbb{R}^{D\times D})^T &\to (\mathbb{R}^{D\times D})^{T-m} & P_{Rm}\colon (\mathbb{R}^{D\times D})^T &\to (\mathbb{R}^{D\times D})^{T-m} \\ \boldsymbol{A} &\mapsto (A_1,\dots,A_{T-m}) & \boldsymbol{A} &\mapsto (A_{m+1},\dots,A_T) \end{aligned} \tag{43}$$

with $m \in [1, T-1]$.

In order to decouple the involved matrices, we define two dual variables $\boldsymbol{Z}_0$ and $\boldsymbol{Z} = (\boldsymbol{Z}_L, \boldsymbol{Z}_R)$, where $\boldsymbol{Z}_L = (\boldsymbol{Z}_{Lm})_{1\le m\le T-1}$ and $\boldsymbol{Z}_R = (\boldsymbol{Z}_{Rm})_{1\le m\le T-1}$. To ease the notation, let $\boldsymbol{Z} = (\boldsymbol{Z}_L, \boldsymbol{Z}_R)$ where $\boldsymbol{Z}_L = (\boldsymbol{Z}_{Lm})_{1\le m\le T-1}$ and $\boldsymbol{Z}_R = (\boldsymbol{Z}_{Rm})_{1\le m\le T-1}$. The general problem (30) becomes:

$$\left.\begin{aligned} \underset{(\boldsymbol{K},\boldsymbol{Z}_0,\boldsymbol{Z})}{\text{minimize}} \quad & \sum_{t=1}^{T}\Big[-\ell(K_t|X_t) + \alpha\|K_t\|_{od,1}\Big] + \\ & \sum_{m=1}^{T-1}\sum_{t=1}^{T-m}\Big[\kappa_{mt}^{\Psi}\Psi(Z_{Rmt} - Z_{Lmt})\Big] \\ \text{s.t. } & \boldsymbol{Z}_0 = \boldsymbol{K}, \boldsymbol{Z}_{Lm} = P_{Lm}\boldsymbol{K}, \boldsymbol{Z}_{Rm} = P_{Rm}\boldsymbol{K}. \end{aligned}\right\}$$

The Lagrangian then becomes:

$$\begin{aligned}\mathcal{L}_\rho(\boldsymbol{K},\boldsymbol{Z},\boldsymbol{U}) = \sum_{t=1}^{T}\Big[&- \ell(K_t|X_t) + \alpha \left\|Z_{0,t}\right\|_{\text{od},1}\Big] \\ &+ \sum_{m=1}^{T-1}\sum_{t=1}^{T-m} \kappa_{mt}\Psi(Z_{Rmt} - Z_{Lmt}) \\ &+ \frac{\rho}{2}\sum_{t=1}^{T}\left[\|K_t - Z_{0,t} + U_{0,t}\|_F^2 - \|U_{0,t}\|_F^2\right] \\ &+ \frac{\rho}{2}\sum_{m=1}^{T-1}\sum_{t=1}^{T-m}\Big[\|K_t - Z_{Lmt} + U_{Lmt}\|_F^2 - \|U_{Lmt}\|_F^2 \\ &\qquad + \|K_{t+m} - Z_{Rmt} + U_{Rmt}\|_F^2 - \|U_{Rmt}\|_F^2\Big]\end{aligned} \tag{44}$$

Then, the ADMM algorithm for problem (44) writes down as

Algorithm 6 ADMM algorithm for the minimisation of TGL_κ, TIGM_κ and TPGM_κ

1: **for** $k = 1, \dots$ **do**
2: $\quad \boldsymbol{K}^{\iota+1} = \underset{\boldsymbol{K}}{\text{argmin}}\ \mathcal{L}_\rho(\boldsymbol{K}, \boldsymbol{Z}^\iota, \boldsymbol{U}_0^\iota, \boldsymbol{U}^\iota)$
3: $\quad \boldsymbol{Z_0}^{\iota+1} = \underset{\boldsymbol{Z_0}}{\text{argmin}}\ \mathcal{L}_\rho(\boldsymbol{K}^{\iota+1}, \boldsymbol{Z}^\iota, \boldsymbol{U}_0^\iota, \boldsymbol{U}^\iota)$
4: $\quad \boldsymbol{Z}^{\iota+1} = \begin{bmatrix} \boldsymbol{Z}_L^{\iota+1} \\ \boldsymbol{Z}_R^{\iota+1} \end{bmatrix} = \underset{\boldsymbol{Z}}{\text{argmin}}\ \mathcal{L}_\rho(\boldsymbol{K}^{\iota+1}, \boldsymbol{Z}, \boldsymbol{U}_0^\iota, \boldsymbol{U}^\iota)$
5: $\quad \boldsymbol{U}_0^{\iota+1} = \boldsymbol{U}_0^\iota + \left[K^{\iota+1} - Z_0^{\iota+1}\right]$
6: $\quad \boldsymbol{U}^{\iota+1} = \begin{bmatrix} \boldsymbol{\Upsilon}_L^k \\ \boldsymbol{\Upsilon}_R^k \end{bmatrix} + \begin{bmatrix} P_L\boldsymbol{K}^{\iota+1} - \boldsymbol{Z}_L^{\iota+1} \\ P_R\boldsymbol{K}^{\iota+1} - \boldsymbol{Z}_R^{\iota+1} \end{bmatrix}$

before, that is the iterative optimisation of $\boldsymbol{K}, \boldsymbol{Z}, \boldsymbol{U}$, respectively.

A.1 K and Z_0 step

The computation of $\boldsymbol{K}$ and $\boldsymbol{Z_0}$ can be performed separately for each t. We put this two steps together as, for the Ising and Poisson temporal models we solve them in a unique minimisation procedure while for the Guassian case, as they

allowed separate closed-form solution we solve them separately. The K step at tie t is defined as

$$\begin{aligned} K_t^{\iota+1} &= \underset{K}{\operatorname{argmin}} \ \ell(K_t|X_t) + \frac{\rho}{2N_t}\Bigg[\|K - Z_{0,t} + U_{0,t}\|_F^2 \\ &\quad + \sum_{m=1}^{T-1}\Bigg(\delta_{t\leq T-m}\|K - Z_{Lmt} + U_{Lmt}\|_F^2 \\ &\qquad + \delta_{t\geq m+1}\|K - Z_{Rmt} + U_{Rmt}\|_F^2\Bigg)\Bigg] \\ &= \underset{K}{\operatorname{argmin}} \ \ell(K_t|X_t) + \frac{T\rho}{2N_t}\|K - A_t\|_F^2 \end{aligned} \tag{45}$$

with

$$\begin{aligned} A_t = \frac{1}{T}\Bigg[&Z_{0,t} - U_{0,t} + \\ &\sum_{m=1}^{T-1}\delta_{t\leq T-m}(Z_{Lmt} - U_{Lmt}) + \delta_{t\geq m+1}(Z_{Rmt} - U_{Rmt})\Bigg] \end{aligned}$$

since $1 + \sum_{m=1}^{T-1}(\delta_{t\leq T-m} + \delta_{t\geq m+1}) = T$. Note that the last equality in (45) follows from the symmetry of K.
Problem (45) needs to be solved differently depending on the type of likelihood. If the likelihood is Gaussian it can be solved in closed form, if the likelihood is either Ising or Poisson it requires a nested minimisation algorithm.
The Z_0 step at time t is defined as

$$\begin{aligned} Z_{0,t}^{\iota+1} &= \underset{Z}{\operatorname{argmin}} \ \alpha\|Z\|_{od,1} + \frac{\rho}{2}\|K_t - Z + U_{0,t}\|_F^2 \\ &= \underset{Z}{\operatorname{argmin}} \ \alpha\|Z\|_{od,1} + \frac{\rho}{2}\|Z - B_t\|_F^2 \end{aligned}$$

with $B_t = K_t + U_{0,t}$.

A.1.1 *Gaussian*

Here we are solving the problem presented in Equation (34), *i.e.*we are instantiating the likelihood with ℓ_{GGM}. Note that, with this likelihood the symmetry of K also guarantees the log det to be well-defined. In this case problem (45) can be explicitly solved. Indeed, Fermat's rule yields:

$$C_t - \frac{\rho}{N_t}\frac{A_t^\iota + A_t^{\iota\top}}{2} = K^{-1} - \frac{T\rho}{N_t}K. \tag{46}$$

Then the solution to Equation (46) is

$$K_t^{\iota+1} = \frac{N_t}{2T\rho}V^\iota\left(-E^\iota + \sqrt{(E^\iota)^2 + \frac{4T\rho}{N_t}I}\right)V^{k\top}$$

where $V^\iota E^\iota V^{\iota\top}$ is the eigenvalue decomposition of $C_t - \frac{T\rho}{N_t}\frac{A_t^\iota + A_t^{\iota\top}}{2}$.

The solution for $Z_{0,t}^{\iota+1}$ can still be performed in closed form as

$$Z_{0,t}^{\iota+1} = \text{soft-thresholding}(K_t + U_{0,t}, \frac{\alpha}{\rho}).$$

A.1.2 *Ising*

Here we are solving the problem presented in Equation (35), *i.e.*, we are instantiating the likelihood with ℓ_{IGM}. For each K_t we have to solve D neighbourhood selection problems, indeed the problem cannot be solved in closed form. Given the constraint $K_t \in \mathcal{S}^D$, *i.e.*, the inferred adjacency matrix needs to be symmetrical we nest the neighbourhood selection problems in a global minimisation procedure in order to guarantee such constraint. Differently from the Gaussian case, here we unify the K and Z_0 step, and therefore we use a Forward-Backward splitting (FBS) algorithm (Combettes and Wajs, 2005), an iterative procedure that computes the gradient on the differentiable part of the functional and then use a proximal operator on the non-differentiable part. Given the functional in Equation (35) we define for each time t and each variable v

$$\begin{aligned} f(K_t[v,:]) =& \log\{\exp(X_t K_t[v,:]^\top) \\ &+ \exp(-X_t K_t[v,:]^\top)\} - \frac{1}{N_t}(X_t[:,v]^\top X_t K_t[v,:]^\top)\} \\ &+ \frac{T\rho}{2}\|K_t[v,:] - A_t[v,:]d\|_F^2 \end{aligned}$$

with

$$A_t = \frac{1}{T}\sum_{m=1}^{T-1} \delta_{t\leq T-m}(Z_{Lmt} - U_{Lmt}) + \delta_{t\geq m+1}(Z_{Rmt} - U_{Rmt})$$

and

$$g(K_t[v,:]) = \alpha\|K_t[v,:]]\|_1$$

Then the minimisation for each K_t is performed as in Algorithm 7 where the gradient of the function f can be computed for each variable v given the decomposability of the function as:

$$\begin{aligned} \nabla_{K_t[v,:]} f =& X_t[:,\setminus v]\left(\frac{\exp(X_t K_t[v,:]^\top) - \exp(-X_t K_t[v,:]^\top}{\exp(X_t K_t[v,:]^\top) + \exp(-X_t K_t[v,:]^\top)} - \frac{1}{N_t} X_t[:,v]\right) \\ &+ T\rho(K_t[v,:] - A_t[v,:]) \end{aligned}$$

Algorithm 7 K step for Temporal Ising model

$\gamma = 0.001$
$\epsilon = 0.001$
$\boldsymbol{K} = \text{zeros}(T, D, D)$
$\boldsymbol{G} = \text{zeros}(T, D, D)$
for $\iota = 1, \dots$ **do**
  for $v = 1, \dots, D$ **do**
    for $t = 1, \dots, T$ **do**
      $\boldsymbol{G}_t[v, :] = \nabla_{K_t[v,:]} f$
  $\boldsymbol{K}^\iota = \boldsymbol{K}^{\iota-1} - \gamma \boldsymbol{G}$
$\boldsymbol{K}^\iota = \text{soft-thresholding}(\boldsymbol{K}^\iota, \gamma\alpha)$
if $\|\boldsymbol{K}^\iota - \boldsymbol{K}^{\iota-1}\|_2^2 < \epsilon$ **then**
  break
return $\boldsymbol{K}^\iota$

The setting $\gamma = 0.001$ has been decided after empirical validation of the values.

A.1.3 *Poisson*

Here, we are solving the problem presented in Equation (36), *i.e.*we are instantiating the likelihood with ℓ_{PGM}. Again for each K_t we have to solve D neighbourhood selection problems and we solve together the K and Z_0 step, using FBS algorithm (Combettes and Wajs, 2005). We couple such algorithm with a line-search procedure to find the best gradient step (Wan et al., 2016). Given the functional in Equation (36) we define for each time t and each variable v

$$f(K_t[v,:]) = \ell_{PGM}(X_t|K_t[v,:]) + \frac{T\rho}{2}\|K_t[v,:] - A_t[v,:]\|_F^2$$

with

$$A_t = \frac{1}{T}\sum_{m=1}^{T-1} \delta_{t\leq T-m}(Z_{Lmt} - U_{Lmt}) + \delta_{t\geq m+1}(Z_{Rmt} - U_{Rmt})$$

and

$$g(K_t[v,:]) = \alpha\|K_t[v,:]]\|_1$$

the minimisation algorithm for each time t is presented in Algorithm 8 where the gradient of f in variable v is defined as

$$\begin{aligned}\nabla_{K_t[v,:]} f = &-\frac{1}{N_t}\sum_{i=1}^{N_t}\Big[X_t[i,v]X_t[i,\setminus v] - X_t[i,\setminus v]\exp(X_t[i,\setminus v]K_t^\top[v,\setminus v])\Big] \\ &+ T\rho(K_t[v,:] - A_t[v,:])\end{aligned}$$

Algorithm 8 K step for Temporal Poisson model

$\gamma = 1$
$\epsilon = 0.001$
$\boldsymbol{K} = \text{zeros}(T, D, D)$
$\boldsymbol{G} = \text{zeros}(T, D, D)$
for $\iota = 1, \dots$ **do**
  for $t = 1, \dots, T$ **do**
    $\boldsymbol{G}_t[v, :] = \nabla_{K_t[v,:]} f$
  while $f(K_t^\iota) > f(K_t^{\iota-1}) - G_t(K_t^\iota - K_t^{\iota-1}) + \frac{1}{2\gamma}\|K_t^\iota - K_t^{\iota-1}\|_F^2$ **do**
    $\gamma = \gamma/2$
    $K_t^\iota = K_t^{\iota-1} - \gamma G_t$
    $K_t^\iota = \text{soft-thresholding}(K_t^\iota, \gamma\alpha)$
if $\|K_t^\iota - K_t^{\iota-1}\|_2^2 < \epsilon$ **then**
  break
return K_t^ι

A.2 Zs step

Variables

$$\boldsymbol{Z}_L = (\boldsymbol{Z}_{Lmt})_{1\le m\le T-1, 1\le t\le T}$$
$$\boldsymbol{Z}_R = (\boldsymbol{Z}_{Rmt})_{1\le m\le T-1, 1\le t\le T}$$

are easily separable. Hence, the minimisation on them can be applied to their single components.

$$\begin{aligned}\begin{bmatrix} Z_{Lmt}^{\iota+1} \\ Z_{Rmt}^{\iota+1} \end{bmatrix} = \underset{Z_L, Z_R}{\text{argmin}} \;\; & \kappa_{mt}^{\Psi}\, \Psi(Z_R - Z_L) \\ & + \frac{\rho}{2}\Big[\, \|K_t^\iota - Z_L + Y_{Lmt}^\iota\|_F^2 + \|K_{t+m}^\iota - Z_R + Y_{Rmt}^\iota\|_F^2 \,\Big].\end{aligned} \tag{47}$$

Let $\hat{\Psi}\begin{bmatrix} Z_L \\ Z_R \end{bmatrix} = \Psi(Z_R - Z_L)$. Then, Problem (47) can be solved with an unique update:

$$\begin{bmatrix} Z_{Lmt}^{\iota+1} \\ Z_{Rmt}^{\iota+1} \end{bmatrix} = \text{prox}_{\frac{\kappa_{mt}^{\Psi}}{\rho}\hat{\Psi}(\cdot)}\left(\begin{bmatrix} K_t^\iota + Y_{Lmt}^\iota \\ K_{t+m}^\iota + Y_{Rmt}^\iota \end{bmatrix}\right).$$

The solution of the proximal operators of the different Ψ functions are available in (Hallac et al., 2017a).

A.3 Termination Criterion

The termination criterion is based on the primal and dual residuals $\|r^\iota\|_2^2 \leq \epsilon^{\text{pri}}$ and $\|s^\iota\|_2^2 \leq \epsilon^{\text{dual}}$ (Boyd et al., 2011). At each iteration ι such values are computed as:

$$\begin{aligned}
\|r^\iota\|_2^2 &= \|\boldsymbol{K}^\iota - \boldsymbol{Z}_0^\iota\|_F^2 + \|P_L\boldsymbol{K}^\iota - \boldsymbol{Z}_L^\iota\|_F^2 + \|P_R\boldsymbol{K}^\iota - \boldsymbol{Z}_R^\iota\|_F^2 \\
\|s^\iota\|_2^2 &= \rho\Big(\left\|\boldsymbol{Z}_0^\iota - \boldsymbol{Z}_0^{k-1}\right\|_F^2 + \left\|\boldsymbol{Z}_L^\iota - \boldsymbol{Z}_L^{k-1}\right\|_F^2 + \left\|\boldsymbol{Z}_R^\iota - \boldsymbol{Z}_R^{k-1}\right\|_F^2 \Big) \\
\epsilon^{\text{pri}} &= c + \epsilon^{\text{rel}}\ \max\left(D_1^\iota, D_2^\iota\right) \\
\epsilon^{\text{dual}} &= c + \epsilon^{\text{rel}}\rho\left(D_3^\iota\right)
\end{aligned}$$

where $c = \epsilon^{\text{abs}} dT$, ϵ^{abs} and ϵ^{rel} are arbitrary tolerance parameters, $\|D_1^\iota\|_F^2 = \|\boldsymbol{Z}_0^\iota\|_F^2 + \|\boldsymbol{Z}_L^\iota\|_F^2 + \|\boldsymbol{Z}_R^\iota\|_F^2$, $\|D_2^\iota\|_F^2 = \|\boldsymbol{K}^\iota\|_F^2 + \|P_L\boldsymbol{K}^\iota\|_F^2 + \|P_R\boldsymbol{K}^\iota\|_F^2$ and $\|D_3^\iota\|_F^2 = \|\boldsymbol{U}_0^\iota\|_F^2 + \|\boldsymbol{U}_L^\iota\|_F^2 + \|\boldsymbol{U}_R^\iota\|_F^2$.

A.4 The problem is separable

Particular instantiation of the problems above, in particular when we select $\Psi = \ell_1$ or $\Psi = \ell_2^2$ makes the problem separable, in such a way to consider the minimisation of a single variable in time ignoring the behaviour of the other when solving the problem. This is particularly indicated in contexts like Bernoulli or Poisson where the minimisation is naturally done in a single variable and can be extended to kernel inclusion easily. In (Tomasi et al., 2018a) we studied this case for the TGL problem proposing a solution based on Forward-Backward Splitting (FBS) that relied on line searches for the parameters of the algorithm and relax the assumptions, so to include the type of problems we are interested in, while maintaining strong theoretical convergence guarantees (Salzo, 2017). We do not want to go into many technical details but we show some experimental comparison that show how FBS can be a valuable alternative to ADMM.

A.4.1 *Experiments*

The performance of the proposed methods has been assessed on synthetic data in terms of the number of iterations, execution time, and space scalability.
In particular, we compared the two version of the FBS based algorithm FBS-LS(γ) and FBS-LS(γ, λ) that are different depending on how the line search is performed, with the ADMM algorithm proposed in (Hallac et al., 2017a).

CONVERGENCE Data were generated starting from a set of precision matrices $\mathbf{K} = (K_1, \ldots, K_T)$, related in time according to a specific behaviour while guaranteeing that $K_t \in \mathcal{S}_{++}^D$ for $t = 1, \ldots, T$. In particular, we generated two

	precision	0.1		0.01		0.001	
	score	iter.	time [s]	iter.	time [s]	iter.	time [s]
ℓ_1	FBS-LS(γ)	22 ± 1	4.8 ± 0.7	24 ± 1	5.1 ± 0.4	26 ± 1	5.0 ± 0.3
	FBS-LS(γ,λ)	22 ± 1	4.6 ± 0.7	24 ± 1	5.4 ± 0.5	26 ± 1	5.6 ± 0.5
	ADMM	1060 ± 553	75.2 ± 39.8	2623 ± 1757	184.3 ± 122.3	4312 ± 1536	301.6 ± 107.4
ℓ_2^2	FBS-LS(γ)	72 ± 19	7.9 ± 2.7	107 ± 30	11.7 ± 4.1	137 ± 40	14.9 ± 5.5
	FBS-LS(γ, λ)	72 ± 19	8.3 ± 2.8	104 ± 31	12.1 ± 4.5	129 ± 41	14.9 ± 6.0
	ADMM	192 ± 24	13.2 ± 1.7	252 ± 42	17.3 ± 3.1	453 ± 66	30.4 ± 3.8

TABLE 13. *Average results in terms of average number of iteration and CPU time in seconds for the comparison of the minimisation of the Time-Varying Graphical Lasso with the FBS with two different types of line search and the ADMM across different runs and for two types of temporal consistency functions. Results are shown as the accuracy in approximating the solution increases.*

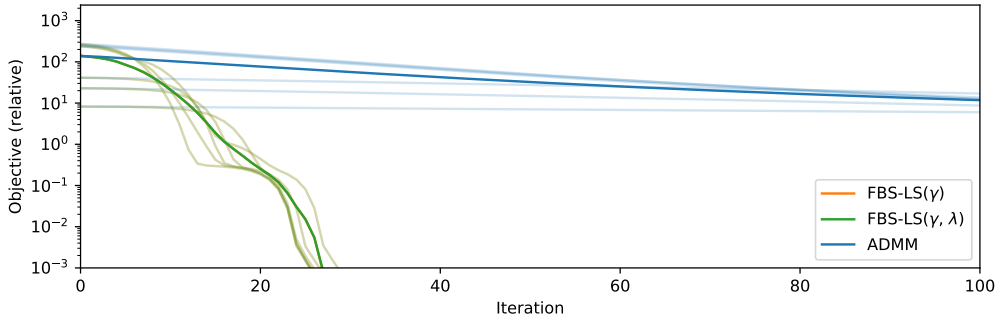
data sets according to different temporal behaviours, consisting of $N_t = 200$ samples in $\mathbb{R}^D$ with $D = 200$ and $T = 10$ time stamps. The first data set was obtained by modelling the interactions between variables across time according to a square waveform. Under such schema, the interactions may be zero or positive at particular time points, but the transition between those states is non-smooth. The second data set is generated modelling variable interactions according to a smooth sinusoidal behaviour. Hence, the interactions were constrained to change slowly in time.

We considered the time-varying graphical lasso with the two temporal penalties (TGL-ℓ_1) and (TGL-ℓ_2^2), according to the type of the data set. As for the hyperparameters (α, β), we considered the search space $[0.1, 1] \times [0.1, 5]$ for (TGL-ℓ_1) and $[0.1, 1] \times [0.01, 0.1]$ for (TGL-ℓ_2^2). We performed a Bayesian optimisation procedure, and we checked that the best hyper-parameters lie in the interior of the search space (do not belong to the boundary). In particular, $(\alpha^*, \beta^*) = (0.111, 4.855)$ for (TGL-ℓ_1), while $(\alpha^*, \beta^*) = (0.789, 0.020)$ for (TGL-ℓ_2^2). Then, we set a grid on the search space and ran the two proposed algorithms FBS-LS(γ) and FBS-LS(γ, λ) as well as ADMM, for the corresponding values of the hyper-parameters.

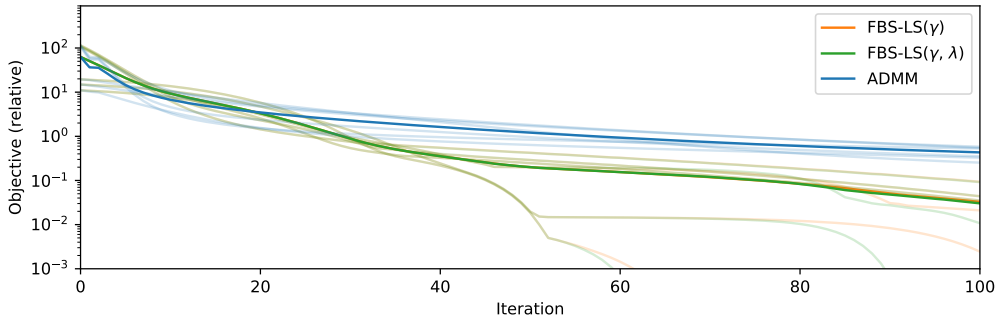
We evaluated the performance of the proposed methods with respect to the ground truth. We computed the mean squared error (MSE) for each algorithm after convergence. The achieved MSE was the same for each algorithm ($0.648 \cdot 10^{-4}$ for (TGL-ℓ_1), $0.498 \cdot 10^{-4}$ for (TGL-ℓ_2^2)).

Table 13 reports the performance of the three algorithms across the different runs in terms of the number of iterations and CPU times for achieving a given precision. For each pair of hyper-parameters, the minimum m_* is estimated as the best value obtained in 500 iterations among the different algorithms.

In this experiment, both FBS-based algorithms clearly outperform the ADMM. FBS-based algorithms are able, in only a few iterations, to increase the precision of order of magnitudes, for both ℓ_1 and ℓ_2^2 set of experiments. We note, however, that the difference in the convergence behaviour with respect to



(A) ℓ_1-*norm temporal penalty.*



(B) ℓ_2^2-*norm temporal penalty.*

FIGURE 48. *Representation of the relative objective value of the FBS with two different types of line search procedures and the ADMM as iterations increase.*

ADMM is less substantial in the case of ℓ_2^2. In the case of ℓ_1, FBS has a higher cost per iteration with respect to ADMM. This is due to the computation of the proximity operator of the fused lasso penalty. In the case of ℓ_2^2, instead, the cost is lower because the proximity operator of the nonsmooth (penalty) term simplifies to a soft-thresholding. Finally, for (TGL-ℓ_2^2), we point out the better performance of FBS-LS(γ, λ) against FBS-LS(γ).
Figure 48 shows the relative objective value across the first 100 iterations and multiple runs for FBS-based algorithms and ADMM.The relative value is obtained as $\frac{|\text{obj}_\iota - m_*|}{|m_*|}$, where m_* is the minimum objective value obtained across 500 iterations, and obj_k is the value of the objective function at iteration ι. The averaged value is depicted in bold line. In particular, in the case of the (TGL-ℓ_1), FBS-based algorithms clearly surpass the ADMM in terms of convergence rate (Figure 48a). We note that the two algorithms FBS-LS(γ) and FBS-LS(γ, λ) completely overlap in the case of (TGL-ℓ_1), whereas FBS-LS(γ, λ) shows to convergence slightly faster than FBS-LS(γ). The poor convergence rate of ADMM may be due to the need of reaching a consensus among a large number of variables which a typical scenario in the inference of time-varying networks.

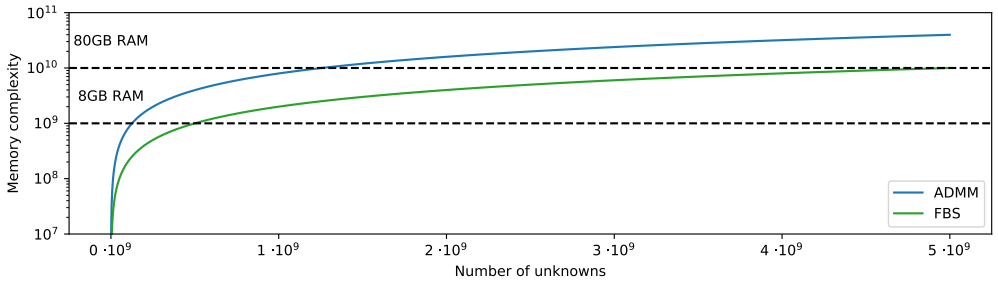


FIGURE 49. *Memory requirement for FBS and ADMM minimisation algorithms for the Time-Varying Graphical Lasso (TVGL) as the number of unknowns increases keeping T fixed to 50 and letting D vary. The matrices are stored in memory in double precision.*

A.4.2 *Scalability*

FBS-based and ADMM-based optimisations feature different memory requirements. In particular, FBS-based implementation requires $O(2D^2T)$ in space, for keeping in memory both the precision and empirical covariance matrices at all time points. Instead, ADMM-based implementation requires more variables due to the consensus framework and the presence of dual variables. More specifically, in our setting, it requires $O(4D^2(2T-1))$ space complexity (Hallac et al., 2017a). The difference between the two complexities consists in a multiplicative factor which, however, may have an impact in the analysis of large data sets.
Figure 49 shows the difference in space complexity as the number of unknowns ($TS(S+1)/2$) of the problem grows. We note that such computations do not take into account the use of optimised data structures for sparse data. Better performance may be achieved by exploiting the structure and the sparsity of the involved matrices, but we leave such investigation for future work.

A.5 Summary

In this appendix we presented the minimisation algorithms for TGL_κ, TIGM_κ, TPGM_κ based on ADMM optimisation algorithm. We also presented a brief overview of an alternative minimisation approach in the Gaussian case that can be used in presence of specific temporal consistency functions. This may be particularly suited for Ising and Poisson problems that work on single variable neighbourhood estimation and it is worth to further investigate.

B

Minimisation LTGL$_\kappa$

Problem (40) is convex, provided that the penalty functions Ψ and Φ are convex, and it is coercive because of the regularisers, so it admits (global) solutions Tomasi et al., 2018b. Nonetheless, its optimisation is challenging in practice due to the high number of unknown matrices involved ($2T$, for a total of $2T\frac{d(d+1)}{2}$ unknowns of the problem). In line with the recent advances of graphical models Danaher, Wang and Witten, 2014; Hallac et al., 2017a; Ma, Xue and Zou, 2013; Tomasi et al., 2018b we recur to the ADMM minimisation algorithmM (Boyd et al., 2011) similarly to what we did for TGL$_\kappa$. The most computationally expensive task performed by our solver is represented by two eigenvalue decompositions, with a complexity of $O(D^3)$, to solve both $\mathbf{R}$ and $\mathbf{L}$ steps.
In order to decouple the involved matrices, we define three dual variables $\boldsymbol{R}$, $\boldsymbol{Z} = (\boldsymbol{Z}_1, \boldsymbol{Z}_2)$ and $\boldsymbol{W} = (\boldsymbol{W}_1, \boldsymbol{W}_2)$. Given the two projections in Equation (43) with $m \in [1, T-1]$. Problem (40) becomes:

$$\left.\begin{aligned} &\underset{\substack{(\boldsymbol{K},\boldsymbol{L},\boldsymbol{R},\boldsymbol{Z},\boldsymbol{W}) \\ \boldsymbol{L}_t \succeq 0}}{\text{minimize}} \sum_{t=1}^{T} \Big[-n_t \, \ell(S_t, R_t) + \alpha \|K_t\|_{od,1} + \tau \|L_t\|_* \Big] \\ &\qquad + \sum_{m=1}^{T-1} \sum_{t=1}^{T-m} \Big[\kappa_{mt}^{\Psi} \Psi(Z_{Rmt} - Z_{Lmt}) + \kappa_{mt}^{\Phi} \Phi(W_{Rmt} - W_{Lmt}) \Big] \\ &\text{s.t. } \boldsymbol{R} = \boldsymbol{K} - \boldsymbol{L}, \boldsymbol{Z}_{Lm} = P_{Lm}\boldsymbol{K}, \boldsymbol{Z}_{Rm} = P_{Rm}\boldsymbol{K}, \\ &\quad \boldsymbol{W}_{Lm} = P_{Lm}\boldsymbol{L}, \boldsymbol{W}_{Rm} = P_{Rm}\boldsymbol{L}. \end{aligned}\right\}$$

To ease the notation, let $\boldsymbol{Z} = (\boldsymbol{Z}_L, \boldsymbol{Z}_R)$ and $\boldsymbol{W} = (\boldsymbol{W}_L, \boldsymbol{W}_R)$, where $\boldsymbol{Z}_L = (\boldsymbol{Z}_{Lm})_{1 \le m \le T-1}$ and $\boldsymbol{Z}_R = (\boldsymbol{Z}_{Rm})_{1 \le m \le T-1}$ (the same applies for $\boldsymbol{W}_L$, $\boldsymbol{W}_R$, $\boldsymbol{P}_L$, $\boldsymbol{P}_R$).

The corresponding augmented Lagrangian is:

$$
\begin{aligned}
&\mathcal{L}_\rho(\boldsymbol{K}, \boldsymbol{L}, \boldsymbol{R}, \boldsymbol{Z}, \boldsymbol{W}, \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{U}) \\
&= \sum_{t=1}^{T} \Big[- n_t \; \ell(S_t, R_t) + \alpha \left\| K_t \right\|_{\mathrm{od},1} + \tau \left\| L_t \right\|_* + \mathbb{I}(L \succeq 0) \Big] \\
&+ \sum_{m=1}^{T-1} \sum_{t=1}^{T-m} \Big[\kappa_{mt}^{\Psi} \Psi(Z_{Rmt} - Z_{Lmt}) + \kappa_{mt}^{\Phi} \Phi(W_{Rmt} - W_{Lmt}) \Big] \\
&+ \frac{\rho}{2} \sum_{t=1}^{T} \Big[\left\| R_t - K_t + L_t + X_t \right\|_F^2 - \left\| X_t \right\|_F^2 \Big] \\
&+ \frac{\rho}{2} \sum_{m=1}^{T-1} \sum_{t=1}^{T-m} \Big[\left\| K_t - Z_{Lmt} + Y_{Lmt} \right\|_F^2 - \left\| Y_{Lmt} \right\|_F^2 \\
&\qquad\qquad + \left\| K_{t+m} - Z_{Rmt} + Y_{Rmt} \right\|_F^2 - \left\| Y_{Rmt} \right\|_F^2 \Big] \\
&+ \frac{\rho}{2} \sum_{m=1}^{T-1} \sum_{t=1}^{T-m} \Big[\left\| L_t - W_{Lmt} + U_{Lmt} \right\|_F^2 - \left\| U_{Lmt} \right\|_F^2 \\
&\qquad\qquad + \left\| L_{t+m} - W_{Rmt} + U_{Rmt} \right\|_F^2 - \left\| U_{Rmt} \right\|_F^2 \Big]
\end{aligned}
\tag{48}
$$

where $\boldsymbol{X}$, $\boldsymbol{Y} = (\boldsymbol{Y}_L, \boldsymbol{Y}_R)$, $\boldsymbol{U} = (\boldsymbol{U}_L, \boldsymbol{U}_R)$ are the scaled dual variables. The related ADMM algorithm for problem (48) writes down as:

Algorithm 9 ADMM algorithm for the minimisation of LTGL$_\kappa$

for $k = 1, \dots$ **do**

$$
\begin{aligned}
\boldsymbol{R}^{\iota+1} &= \operatorname*{argmin}_{\boldsymbol{R}} \; \mathcal{L}_\rho(\boldsymbol{K}^\iota, \boldsymbol{L}^\iota, \boldsymbol{R}, \boldsymbol{Z}^\iota, \boldsymbol{W}^\iota, \boldsymbol{X}^\iota, \boldsymbol{Y}^\iota, \boldsymbol{U}^\iota) \\
\boldsymbol{K}^{\iota+1} &= \operatorname*{argmin}_{\boldsymbol{K}} \; \mathcal{L}_\rho(\boldsymbol{K}, \boldsymbol{L}^\iota, \boldsymbol{R}^{\iota+1}, \boldsymbol{Z}^\iota, \boldsymbol{W}^\iota, \boldsymbol{X}^\iota, \boldsymbol{Y}^\iota, \boldsymbol{U}^\iota) \\
\boldsymbol{L}^{\iota+1} &= \operatorname*{argmin}_{\boldsymbol{L}} \; \mathcal{L}_\rho(\boldsymbol{K}^{\iota+1}, \boldsymbol{L}, \boldsymbol{R}^{\iota+1}, \boldsymbol{Z}^\iota, \boldsymbol{W}^\iota, \boldsymbol{X}^\iota, \boldsymbol{Y}^\iota, \boldsymbol{U}^\iota) \\
\boldsymbol{Z}^{\iota+1} &= \begin{bmatrix} \boldsymbol{Z}_L^{\iota+1} \\ \boldsymbol{Z}_R^{\iota+1} \end{bmatrix} = \operatorname*{argmin}_{\boldsymbol{Z}} \; \mathcal{L}_\rho(\boldsymbol{K}^{\iota+1}, \boldsymbol{L}^{\iota+1}, \boldsymbol{R}^{\iota+1}, \boldsymbol{Z}, \\
&\qquad\qquad \boldsymbol{W}^\iota, \boldsymbol{X}^\iota, \boldsymbol{Y}^\iota, \boldsymbol{U}^\iota) \\
\boldsymbol{W}^{\iota+1} &= \begin{bmatrix} \boldsymbol{W}_L^{\iota+1} \\ \boldsymbol{W}_R^{\iota+1} \end{bmatrix} = \operatorname*{argmin}_{\boldsymbol{W}} \; \mathcal{L}_\rho(\boldsymbol{K}^{\iota+1}, \boldsymbol{L}^{\iota+1}, \boldsymbol{R}^{\iota+1}, \boldsymbol{Z}^{\iota+1}, \\
&\qquad\qquad \boldsymbol{W}, \boldsymbol{X}^\iota, \boldsymbol{Y}^\iota, \boldsymbol{U}^\iota) \\
\boldsymbol{X}^{\iota+1} &= \boldsymbol{X}^\iota + \left[\boldsymbol{R}^{\iota+1} - \boldsymbol{K}^{\iota+1} + \boldsymbol{L}^{\iota+1} \right] \\
\boldsymbol{Y}^{\iota+1} &= \begin{bmatrix} \boldsymbol{Y}_L^k \\ \boldsymbol{Y}_R^k \end{bmatrix} + \begin{bmatrix} P_L \boldsymbol{K}^{\iota+1} - \boldsymbol{Z}_L^{\iota+1} \\ P_R \boldsymbol{K}^{\iota+1} - \boldsymbol{Z}_R^{\iota+1} \end{bmatrix} \\
\boldsymbol{U}^{\iota+1} &= \begin{bmatrix} \boldsymbol{U}_L^k \\ \boldsymbol{U}_R^k \end{bmatrix} + \begin{bmatrix} P_L \boldsymbol{L}^{\iota+1} - \boldsymbol{W}_L^{\iota+1} \\ P_R \boldsymbol{L}^{\iota+1} - \boldsymbol{W}_R^{\iota+1} \end{bmatrix} .
\end{aligned}
$$

B.1 R step

$$\begin{aligned} R_t^{\iota+1} &= \underset{R}{\operatorname{argmin}} \;\; \operatorname{tr}(S_t R) - \log\det(R) + \frac{\rho}{2n_t}\left\|R - K_t^\iota + L_t^\iota + X_t^\iota\right\|_F^2 \\ &= \underset{R}{\operatorname{argmin}} \;\; \operatorname{tr}(S_t R) - \log\det(R) + \frac{\rho}{2n_t}\left\|R - A_t\right\|_F^2 \end{aligned} \tag{49}$$

with $A_t = K_t^\iota - L_t^\iota - X_t^\iota$. Note that the last equality in (49) follows from the symmetry of R — which also guarantees the log det to be well-defined. Equation (49) can be explicitly solved. Indeed, Fermat's rule yields:

$$S_t - \frac{\rho}{n_t}\frac{A_t^\iota + A_t^{k\top}}{2} = R^{-1} - \frac{\rho}{n_t}R. \tag{50}$$

Then the solution to Equation (50) is

$$R_t^{\iota+1} = \frac{n_t}{2\rho}V^\iota\left(-E^\iota + \sqrt{(E^\iota)^2 + \frac{4\rho}{n_t}I}\right)V^{k\top}$$

where $V^\iota E^\iota V^{k\top}$ is the eigenvalue decomposition of $S_t - \frac{\rho}{n_t}\frac{A_t^\iota + A_t^{k\top}}{2}$.

B.2 K step

$$\begin{aligned} K_t^{\iota+1} &= \underset{K}{\operatorname{argmin}} \;\; \alpha\left\|K\right\|_{od,1} + \frac{\rho}{2}\left\|R_t - K + L_t + X_t\right\|_F^2 \\ &\quad + \frac{\rho}{2}\sum_{m=1}^{T-1}\delta_{t\le T-m}\left\|K - Z_{Lmt} + Y_{Lmt}\right\|_F^2 \\ &\quad + \frac{\rho}{2}\sum_{m=1}^{T-1}\delta_{t\ge m+1}\left\|K - Z_{Rmt} + Y_{Rmt}\right\|_F^2 \\ &= \underset{K}{\operatorname{argmin}} \;\; \alpha\left\|K\right\|_{od,1} + \frac{T\rho}{2}\left\|K - B_t\right\|_F^2 \end{aligned}$$

with

$$\begin{aligned} B_t = \frac{1}{T}\Bigg\{ & R_t + L_t + X_t + \sum_{m=1}^{T-1}\delta_{t\le T-m}(Z_{Lmt} - Y_{Lmt}) \\ & + \sum_{m=1}^{T-1}\delta_{t\ge m+1}(Z_{Rmt} - Y_{Rmt})\Bigg\}. \end{aligned}$$

Hence, the solution is

$$K_t^{\iota+1} = \text{soft-thresholdin}_{\frac{\alpha}{T\rho}}(B_t),$$

B.3 L step

$$
\begin{aligned}
L_t^{\iota+1} &= \underset{L}{\operatorname{argmin}} \;\; \tau \operatorname{tr}(L) + \mathbb{I}(L \succeq 0) + \frac{\rho}{2} \left\| R_t^{\iota+1} - K_t^{\iota+1} + L + X_t^{\iota} \right\|_F^2 \\
&\quad + \frac{\rho}{2} \sum_{m=1}^{T-1} \delta_{t \leq T-m} \left\| L - W_{Lmt} + U_{Lmt} \right\|_F^2 \\
&\quad + \frac{\rho}{2} \sum_{m=1}^{T-1} \delta_{t \geq m+1} \left\| L - W_{Rmt} + U_{Rmt} \right\|_F^2 \\
&= \underset{L}{\operatorname{argmin}} \;\; \tau \operatorname{tr}(L) + \mathbb{I}(L \succeq 0) + \frac{T\rho}{2} \left\| L - C_t^{\iota} \right\|_F^2 \\
&= \underset{L}{\operatorname{argmin}} \;\; \tau \operatorname{tr}(L) + \mathbb{I}(L \succeq 0) + \frac{T\rho}{2} \left\| L - \frac{C_t^{\iota} + C_t^{k\top}}{2} \right\|_F^2
\end{aligned}
\tag{51}
$$

where

$$
\begin{aligned}
C_t^{\iota} = \frac{1}{T} \Bigg\{ & R_t + L_t + X_t + \sum_{m=1}^{T-1} \delta_{t \leq T-m} (W_{Lmt} - U_{Lmt}) \\
& + \sum_{m=1}^{T-1} \delta_{t \geq m+1} (W_{Rmt} - U_{Rmt}) \Bigg\}.
\end{aligned}
$$

Note that the last equality in (51) follows from the symmetry of L. The solution to Problem (51) is:

$$
L_t^{\iota+1} = V^{\iota} \tilde{E} V^{k\top},
$$

where $V^{\iota} E^{\iota} V^{k\top}$ is the eigenvalue decomposition of C_t^{ι}, and

$$
\tilde{E}_{jj} = \max \left(E_{jj}^{\iota} - \frac{\tau}{T\rho}, 0 \right).
$$

B.4 Zs and Ws step

Variables in

$$
\begin{aligned}
\mathbf{Z}_L &= (\mathbf{Z}_{Lmt})_{1 \leq m \leq T-1, 1 \leq t \leq T} \\
\mathbf{Z}_R &= (\mathbf{Z}_{Rmt})_{1 \leq m \leq T-1, 1 \leq t \leq T}
\end{aligned}
$$

are easily separable. Hence, the following minimisation may be applied to their single components.

$$
\begin{aligned}
\begin{bmatrix} Z_{Lmt}^{\iota+1} \\ Z_{Rmt}^{\iota+1} \end{bmatrix} = \underset{Z_L, Z_R}{\operatorname{argmin}} \;\; & \kappa_{mt}^{\Psi} \, \Psi(Z_R - Z_L) \\
& + \frac{\rho}{2} \left[\left\| K_t^{\iota} - Z_L + Y_{Lmt}^{\iota} \right\|_F^2 + \left\| K_{t+m}^{\iota} - Z_R + Y_{Rmt}^{\iota} \right\|_F^2 \right].
\end{aligned}
\tag{52}
$$

Let $\hat{\Psi}\begin{bmatrix} Z_L \\ Z_R \end{bmatrix} = \Psi(Z_R - Z_L)$. Then, Problem (52) can be solved with an unique update:

$$\begin{bmatrix} Z_{Lmt}^{\iota+1} \\ Z_{Rmt}^{\iota+1} \end{bmatrix} = \text{prox}_{\frac{\kappa_{mt}^{\Psi}}{\rho}\hat{\Psi}(\cdot)}\left(\begin{bmatrix} K_t^{\iota} + Y_{Lmt}^{\iota} \\ K_{t+m}^{\iota} + Y_{Rmt}^{\iota} \end{bmatrix}\right).$$

The same applies to $\boldsymbol{W}$ variables. For the particular derivation of different proximal operators, see (Hallac et al., 2017a).

B.5 Termination Criterion

The ADMM algorithm is said to converge when the primal and dual residuals are sufficiently small, *i.e.*, At each iteration k such values are computed as:

$$\begin{aligned}
\|r^{\iota}\|_2 = &\ \|\boldsymbol{R}^{\iota} - \boldsymbol{K}^{\iota} + \boldsymbol{L}^{\iota}\|_F^2 + \|P_L\boldsymbol{K}^{\iota} - \boldsymbol{Z}_L^{\iota}\|_F^2 + \|P_R\boldsymbol{K}^{\iota} - \boldsymbol{Z}_R^{\iota}\|_F^2 \\
&+ \|P_L\boldsymbol{L}^{\iota} - \boldsymbol{W}_L^{\iota}\|_F^2 + \|P_R\boldsymbol{L}^{\iota} - \boldsymbol{W}_R^{\iota}\|_F^2 \\
\|s^{\iota}\|_2 = &\ \rho\Big(\left\|\boldsymbol{R}^{\iota} - \boldsymbol{R}^{k-1}\right\|_F^2 + \left\|\boldsymbol{Z}_L^{\iota} - \boldsymbol{Z}_L^{k-1}\right\|_F^2 + \left\|\boldsymbol{Z}_R^{\iota} - \boldsymbol{Z}_R^{k-1}\right\|_F^2 \\
&+ \left\|\boldsymbol{W}_L^{\iota} - \boldsymbol{W}_L^{k-1}\right\|_F^2 + \left\|\boldsymbol{W}_R^{\iota} - \boldsymbol{W}_R^{k-1}\right\|_F^2\Big) \\
\epsilon^{\text{pri}} = &\ c + \epsilon^{\text{rel}}\ \max\left(D_1^{\iota}, D_2^{\iota}\right) \\
\epsilon^{\text{dual}} = &\ c + \epsilon^{\text{rel}}\rho\left(D_3^{\iota}\right)
\end{aligned}$$

where $c = \epsilon^{\text{abs}}D(T(2T-1))^{1/2}$, ϵ^{abs} and ϵ^{rel} are arbitrary tolerance parameters, $\|D_1^{\iota}\|_F^2 = \|\boldsymbol{R}^{\iota}\|_F^2 + \|\boldsymbol{Z}_L^{\iota}\|_F^2 + \|\boldsymbol{Z}_R^{\iota}\|_F^2 + \|\boldsymbol{W}_L^{\iota}\|_F^2 + \|\boldsymbol{W}_R^{\iota}\|_F^2$, $\|D_2^{\iota}\|_F^2 = \|\boldsymbol{K}^{\iota} - \boldsymbol{L}^{\iota}\|_F^2 + \|P_L\boldsymbol{K}^{\iota}\|_F^2 + \|P_R\boldsymbol{K}^{\iota}\|_F^2 + \|P_L\boldsymbol{L}^{\iota}\|_F^2 + \|P_R\boldsymbol{L}^{\iota}\|_F^2$ and $\|D_3^{\iota}\|_F^2 = \|\boldsymbol{X}^{\iota}\|_F^2 + \|\boldsymbol{Y}_L^{\iota}\|_F^2 + \|\boldsymbol{Y}_R^{\iota}\|_F^2 + \|\boldsymbol{U}_L^{\iota}\|_F^2 + \|\boldsymbol{U}_R^{\iota}\|_F^2$.

C

Synthetic data generation

Throughout this thesis we performed a huge variety of synthetic data experiments to assess the ability of the proposed network inference methods of retrieving the true underlying graph. We now describe in details the mechanisms through which we generated the synthetic data evolution in time. All the presented methods are available in the REGAIN library.

C.1 ℓ_1 evolution schema

The evolution behaviour based on ℓ_1 norm implies that at dependent time points t and t' (typically close in time) their distance Ψ computed as $\|K_t - K_{t'}\|_1$ is small, Figure 50 provides a graphical example of evolving network with ℓ_1 evolution schema.
Given D nodes and T time points, to simulate such situation we generate an initial random graph by fixing the maximum degree of each node (the default value is 2). Then, given this initial graph we randomly select an edge between all the possible edge. If such edge already exist we prune it inserting a zero in the related adjacency matrix, if it does not exist we generate a suitable value v depending on the assumed distribution, and we insert it in the adjacency matrix. More synthetically, we randomly remove or add a connection at each time point.
Then, depending on the distribution, we generated N_t nodes per each randomly generated graph.

WITH LATENT VARIABLES When we are in presence of latent variables, in the context of GGMs, we require to pay more attention to the generation of data. Indeed, we need to satisfy the constraints that the matrix L containing the marginalisation of the latent variables is positive semi-definite and that, the perturbed precision matrix obtained with the subtraction $K - L$ is positive definite. Therefore, given $|O|$ observed variables and $|M|$ latent ones, we generated T time points according to the schema previously introduced and then we compute the Schur complement (see Equation (25)) of such matrix and we use the result to generate the N_t samples.

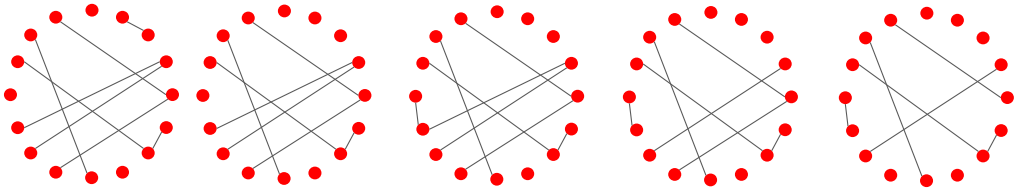


FIGURE 50. *Example of generation of network with ℓ_1 evolution behaviour. Each time has the same structure as the one before plus or minus one edges.*

C.2 ℓ_2^2 evolution schema

The evolution behaviour based on ℓ_2^2 norm implies that at dependent time points t and t' (typically close in time) their distance Ψ computed as $\|K_t - K_{t'}\|_2^2$ is small. Given D nodes and T time points, the first adjacency matrix is generated at random by fixing again the number of degree per each node to 2. In order to comply with the evolving behaviour we add to the initial matrix a random random matrix of small ℓ_2^2 norm in such a way that the differences between two consecutive matrices is small and bounded over time, *i.e.*, $\|K_t - K_{t'}\|_F \leq \epsilon$ for $i = 2, \ldots, T$. The bound ϵ on the norm is chosen *a priori* equal to 0.01.

WITH LATENT VARIABLES In presence of latent variables (in GGMs) we update L_t by maintaining consistency with the theoretical model where $L_t = K_t[OH]K_t[OH]^\top$. Therefore, the update is obtained by adding a random matrix with a small norm to $K_{t'}[OH]$. In this way, the rank of L_t remains the same as the number of latent variables and constant over time.

C.3 Particles diffusion evolution schema

This approach is taken from (Yuan, 2012) and simulates the evolving in time as a diffusion process of particles. Given $|O|$ observed and $|M|$ latent variables and $T = 10$ times, we randomly pick the locations of variables in the space $[0, 1]^2$ where each position (x, y) is sampled from a uniform distribution. Given their positions, we connect a pair of nodes with probability $\psi(\frac{d}{\sqrt{D}})$ where d is the Euclidian distance between the two variables (see Figure 51 for a visual representation). We impose the observed ones to have at most 4 edges and we set their value to 0.245, as for the latent variables we link each of them to $\frac{|M|+|O|}{2}$ variables (both latent and observed) with a value set to $0.98/\frac{|M|+|O|}{2}$ to ensure positive definiteness. The evolution in time is simulated by letting the nodes move in space and updating, for each time point, their probability to be connected. The connections between latent and observed are kept fixed. From the resulting evolved network we sample N_t samples per each time point t.

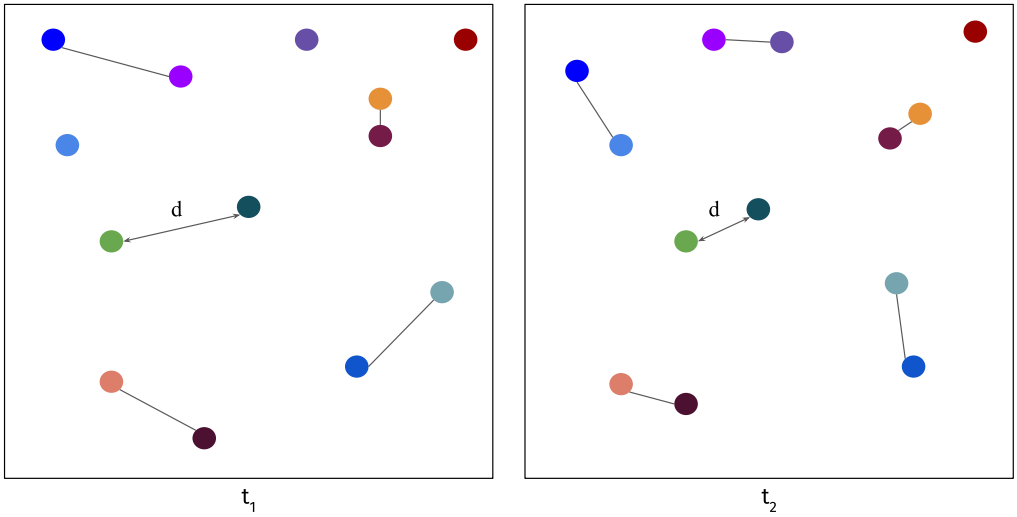


FIGURE 51. *Example of generation of network using particle diffusion approach. We represented two time points t_1 and t_2 in which we let particles (variables) move in space. Such particles are connected with an edge with a certain probability depending on their distance.*

C.4 Cluster-based evolution schema

In this case we want to emulate a situation in which we have recurrences of networks in time with or without a specific repetition pattern. In particular given T time points and the number of possible networks to repeat $k < T$ (that we will call cluster representatives) we randomly select the location of this k networks in time. Note that we can specify if the recurrence should be periodical, for example given networks A, B and C we may want them to repeat as "ABCABC" or "AABBCCAABBCC" or we can specify a completely random type of recurrence. Figure 52 shows a toy example of this evolution schema with 3 cluster representative, 15 time points and periodic recurrence "ABCABC".

Given the positions we generate the networks corresponding to the cluster representatives. Then, suppose one representative is at time t and an other one at time t' such that there is no other representative in the interval $[t, t']$. If $t - t' = d > 0$ we generate d networks that slowly evolve from one representative to an other. The evolving is computed by selecting all the differences (existence or not of an edge) between the two representatives. Let us say that the set of differences has cardinality K, then we impose K/d changes to the middle networks to simulate an evolution. In this way each adjacency matrix is non trivially assigned to a specific cluster.

After we generated such evolution we can sample from the related distribution N_t samples for each time t.

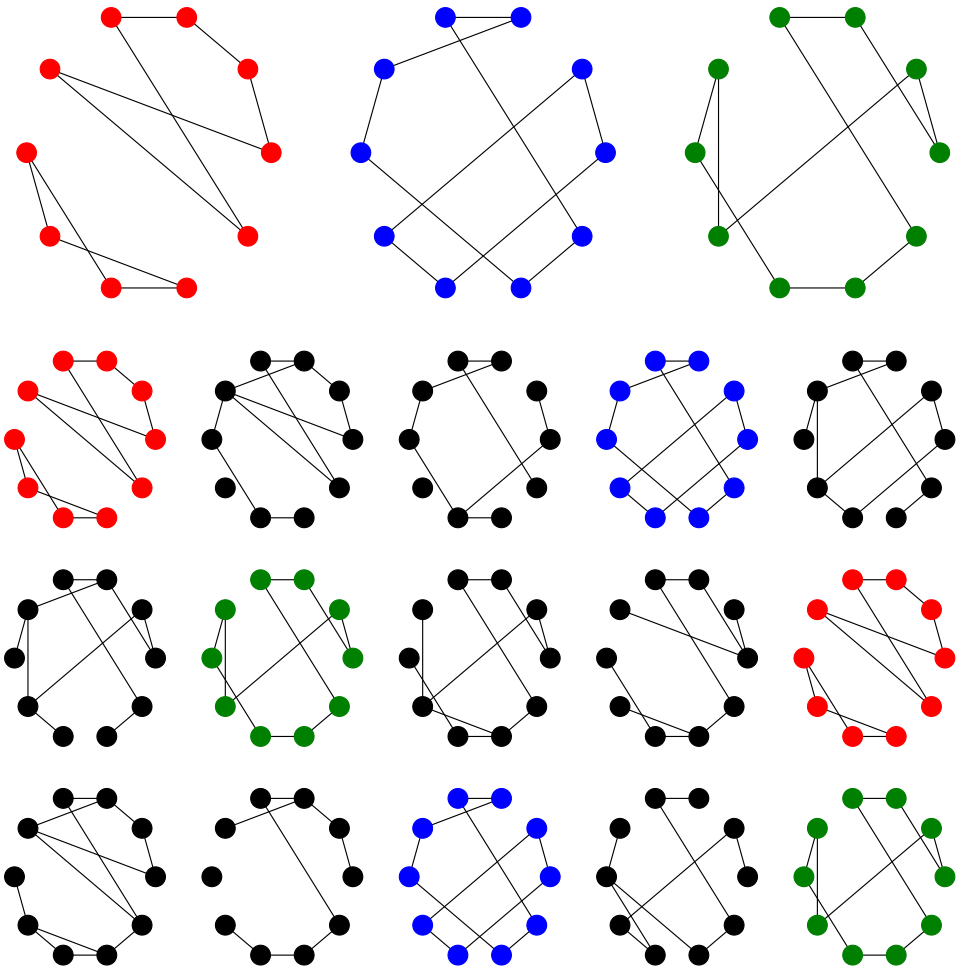


FIGURE 52. *Example of generation of network with repetition in times. In the top row we have three cluster representatives randomly generated. Then, they are periodically positioned in 15 time points. The networks in the middles (black nodes networks) are built by adding or deleting edges in order to smoothly evolve from one representative to an other.*

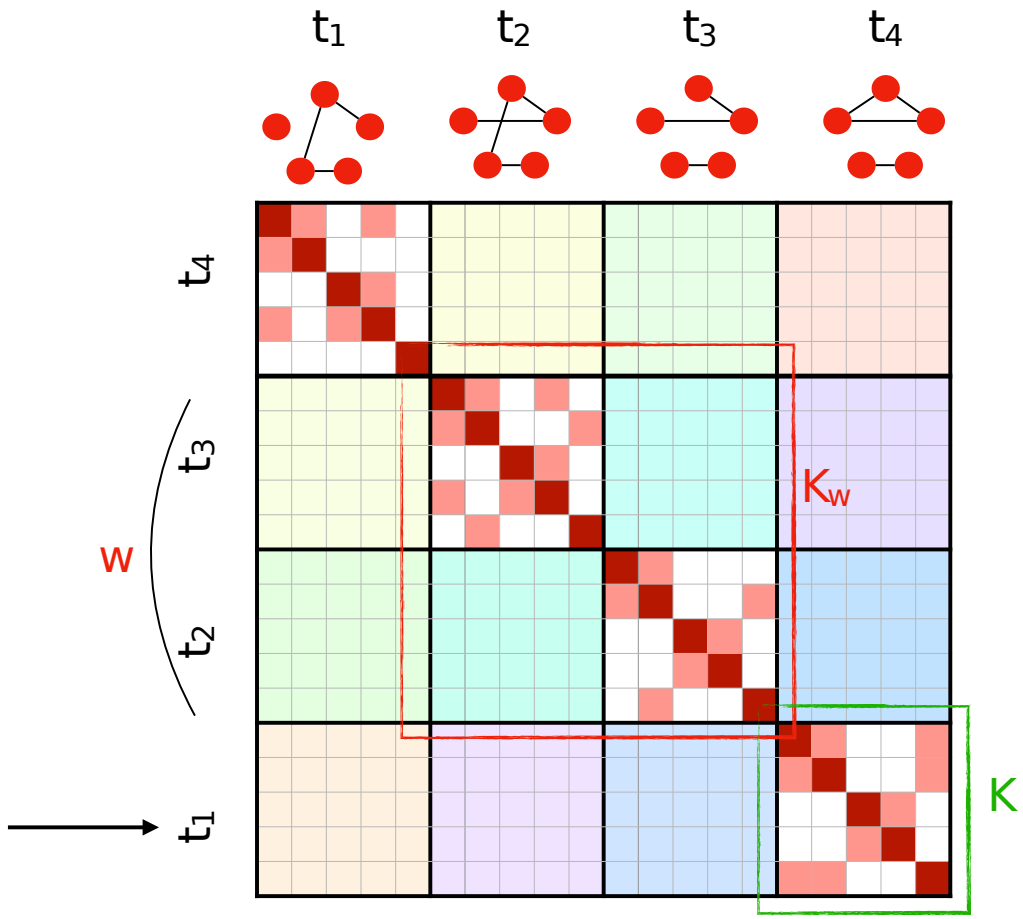


FIGURE 53. *Example of block matrix used to generate conditioned Gaussian samples. On the diagonal block we have the true network structure corresponding to the ones depicted on top. We focus on time t_1 (the block highlighted in green) as we want to sample from its conditioned distribution. In order to simulate temporal dependency we fixed a window of length $w = 2$ that indicates how man previous time points we should consider (in this case $t_)2$ and t_3). We compute the conditioned precision matrix using the Schur complement (see Equation (25)). On the non-diagonal we have symmetric random blocks were the colour indicates a different random generation.*

C.5 Conditioning-based generation

In all data previously generated, we simulated the temporal dependency through a consistency in network structure. Nonetheless, it is possible to further emphasise this dependencies by conditioning each time on the previous ones. This approach is taken from Hallac et al., 2017b and is suitable only in case of GGMs. Consider T times in D variables, we can generate the true adjacency matrices with any schema of the ones previously introduced. Now, let us insert this adjacency matrices in a bigger matrix $\hat{K}$ of dimension $(TD) \times (TD)$ as in Figure 53. Then, fixed a time point t we can condition on the previous w time points were w is an integer number that defines the time window we want to use for perturbing the distribution.

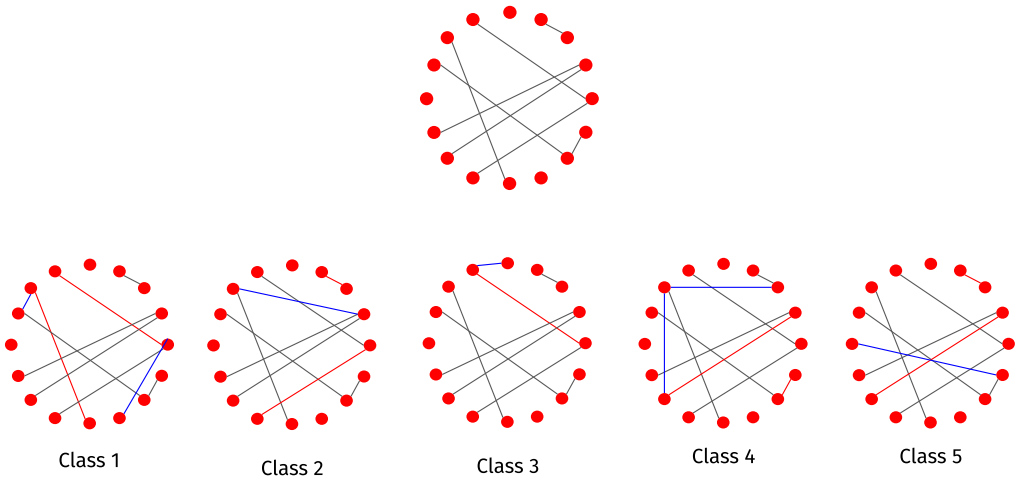


FIGURE 54. *Example of generation of network for multi-class problems. The first row contains the initial adjacency matrix that is perturbed by adding or removing at most 4 edges to obtain 5 adjacency matrices representing 5 different classes.*

Therefore, given the adjacency matrix K_t at time t and the block corresponding to the window w we define the precision matrix of the multivariate normal distribution using the Schur complement. In the indexing we identify the blocks of the matrix $\hat{K}$

$$K(t,w) = \hat{K}[t,t] - \hat{K}[t,t-w:t]\hat{K}[t-w:t,t-w:t]\hat{K}[t-w:t,t]^\top$$

from this matrix $K(t,w)$we sample N_t observations.

C.6 Multi-class schema

The generation of a multi-class network inference problem is similar to the ℓ_1 evolution schema. In particular, given k classes and D variables. We generate an initial random network that represents the basic structure similar across all classes (see Figure 54).
The generation of this initial matrix is performed using the NetworkX package (Hagberg, Swart and S Chult, 2008) either with Erdő-Rényi (Erdős and Rényi, 1960) (with attachment probability $p = 0.2$) or the Barabasi-Albert (Albert and Barabási, 2002) (where each new node is attached with 2 edges to existing nodes) random models. Then, the generation of the k classes is performed by randomly adding or deleting at most 4 edges per each class. We then sample N_k samples from each class distribution.

Bibliography

Abegaz, Fentaw and Ernst Wit (2013). 'Sparse time series chain graphical models for reconstructing genetic networks'. In: *Biostatistics* 14.3, pp. 586–599.

Albert, Réka (2007). 'Network inference, analysis, and modeling in systems biology'. In: *The Plant Cell* 19.11, pp. 3327–3338.

Albert, Réka and Albert-László Barabási (2002). 'Statistical mechanics of complex networks'. In: *Reviews of modern physics* 74.1, p. 47.

Allen, Genevera I and Zhandong Liu (2013). 'A local poisson graphical model for inferring networks from sequencing data'. In: *IEEE transactions on nanobioscience* 12.3, pp. 189–198.

Anandkumar, A., D. Hsu, A. Javanmard and S. Kakade (2013). 'Learning linear bayesian networks with latent variables'. In: *ICML*, pp. 249–257.

Anandkumar, Animashree, Vincent YF Tan, Furong Huang, Alan S Willsky et al. (2012). 'High-dimensional structure estimation in Ising models: Local separation criterion'. In: *The Annals of Statistics* 40.3, pp. 1346–1375.

Bai, J. and S. Ng (2006). 'Evaluating latent and observed factors in macroeconomics and finance'. In: *Journal of Econometrics* 131.1-2, pp. 507–537.

Banerjee, Onureena, Laurent El Ghaoui and Alexandre d'Aspremont (2008). 'Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data'. In: *Journal of Machine learning research* 9.Mar, pp. 485–516.

Barabasi, Albert-Laszlo and Zoltan N Oltvai (2004). 'Network biology: understanding the cell's functional organization'. In: *Nature reviews genetics* 5.2, p. 101.

Beal, Matthew .J. and Zoubin Ghahramani (2003). 'The variational Bayesian EM algorithm for incomplete data: with application to scoring graphical model structures'. In: *Bayesian statistics* 7, pp. 453–464.

Beal, Matthew J, Francesco Falciani, Zoubin Ghahramani, Claudia Rangel and David L Wild (2004). 'A Bayesian approach to reconstructing genetic regulatory networks with hidden factors'. In: *Bioinformatics* 21.3, pp. 349–356.

Belilovsky, Eugene, Gaël Varoquaux and Matthew B Blaschko (2016). 'Testing for differences in Gaussian graphical models: applications to brain connectivity'. In: *Advances in Neural Information Processing Systems*, pp. 595–603.

Bergomi, Mattia G, Massimo Ferri, Pietro Vertechi and Lorenzo Zuffi (2019). 'Beyond topological persistence: Starting from networks'. In: *arXiv preprint arXiv:1901.08051*.

Bergstra, James and Yoshua Bengio (2012). 'Random search for hyper-parameter optimization'. In: *Journal of Machine Learning Research* 13.Feb, pp. 281–305.

Bernardo, JM, MJ Bayarri, JO Berger, AP Dawid, D Heckerman, AFM Smith, M West et al. (2003). 'The variational Bayesian EM algorithm for incomplete data: with application to scoring graphical model structures'. In: *Bayesian statistics* 7, pp. 453–464.

Besag, Julian (1974). 'Spatial interaction and the statistical analysis of lattice systems'. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 36.2, pp. 192–225.

Bianco-Martinez, E, N Rubido, Ch G Antonopoulos and MS Baptista (2016). 'Successful network inference from time-series data using mutual information rate'. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 26.4, p. 043102.

Bien, Jacob and Robert J Tibshirani (2011). 'Sparse estimation of a covariance matrix'. In: *Biometrika* 98.4, pp. 807–820.

Bini, Francesca, Alessia Frati, Mercedes Garcia-Gil, Chiara Battistini, Maria Granado, Maria Martinesi, Marco Mainardi, Eleonora Vannini, Federico Luzzati, Matteo Caleo et al. (2012). 'New signalling pathway involved in the anti-proliferative action of vitamin D3 and its analogues in human neuroblastoma cells. A role for ceramide kinase'. In: *Neuropharmacology* 63.4, pp. 524–537.

Blei, David M, Alp Kucukelbir and Jon D McAuliffe (2017). 'Variational inference: A review for statisticians'. In: *Journal of the American statistical Association* 112.518, pp. 859–877.

Bogdan, Małgorzata, Jayanta K Ghosh and RW Doerge (2004). 'Modifying the Schwarz Bayesian information criterion to locate multiple interacting quantitative trait loci'. In: *Genetics* 167.2, pp. 989–999.

Borriello, Lucia, Robert C Seeger, Shahab Asgharzadeh and Yves A DeClerck (2016). 'More than the genes, the tumor microenvironment in neuroblastoma'. In: *Cancer letters* 380.1, pp. 304–314.

Boyd, Stephen, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein et al. (2011). 'Distributed optimization and statistical learning via the alternating direction method of multipliers'. In: *Foundations and Trends® in Machine learning* 3.1, pp. 1–122.

Bresler, Guy (2015). 'Efficiently learning Ising models on arbitrary graphs'. In: *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*. ACM, pp. 771–782.

Broman, Karl W and Terence P Speed (2002). 'A model selection approach for the identification of quantitative trait loci in experimental crosses'. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 64.4, pp. 641–656.

Capasso, Mario, Sharon J Diskin, Flora Cimmino, Giovanni Acierno, Francesca Totaro, Giuseppe Petrosino, Lucia Pezone, Maura Diamond, Lee McDaniel, Hakon Hakonarson et al. (2014). 'Common genetic variants in NEFL influence gene expression and neuroblastoma risk'. In: *Cancer research*, canres–0431.

Chandrasekaran, Venkat, Pablo A Parrilo and Alan S Willsky (2010). 'Latent variable graphical model selection via convex optimization'. In: *2010 48th*

Annual Allerton Conference on Communication, Control, and Computing (Allerton). IEEE, pp. 1610–1613.

Chandrasekaran, Venkat, Sujay Sanghavi, Pablo A Parrilo and Alan S Willsky (2011). 'Rank-sparsity incoherence for matrix decomposition'. In: *SIAM Journal on Optimization* 21.2, pp. 572–596.

Chang, Andersen, Tianyi Yao and Genevera I Allen (2019). 'Graphical Models and Dynamic Latent Factors for Modeling Functional Brain Connectivity'. In: *2019 IEEE Data Science Workshop (DSW)*. IEEE, pp. 57–63.

Changyong, FENG, WANG Hongyue, LU Naiji, CHEN Tian, HE Hua, LU Ying et al. (2014). 'Log-transformation and its implications for data analysis'. In: *Shanghai archives of psychiatry* 26.2, p. 105.

Chen, Jiahua and Zehua Chen (2008). 'Extended Bayesian information criteria for model selection with large model spaces'. In: *Biometrika* 95.3, pp. 759–771.

Chen, Yixin, Lin Meng and Jiawei Zhang (2019). 'Graph neural lasso for dynamic network regression'. In: *arXiv preprint arXiv:1907.11114*.

Chen, Yunxiao, Xiaoou Li, Jingchen Liu and Zhiliang Ying (2016). 'A fused latent and graphical model for multivariate binary data'. In: *arXiv preprint arXiv:1606.08925*.

Cheng, Lulu, Liang Shan and Inyoung Kim (2017). 'Multilevel Gaussian graphical model for multilevel networks'. In: *Journal of Statistical Planning and Inference* 190, pp. 1–14.

Cheng, NC, M Beitsma, A Chan, I den Camp Op, A Westerveld, J Pronk and R Versteeg (1996). 'Lack of class I HLA expression in neuroblastoma is associated with high N-myc expression and hypomethylation due to loss of the MEMO-1 locus.' In: *Oncogene* 13.8, pp. 1737–1744.

Choi, Myung Jin, Venkat Chandrasekaran and Alan S Willsky (2009). 'Gaussian multiresolution models: Exploiting sparse Markov and covariance structure'. In: *IEEE Transactions on Signal Processing* 58.3, pp. 1012–1024.

Choi, Myung Jin, Vincent YF Tan, Animashree Anandkumar and Alan S Willsky (2011). 'Learning latent tree graphical models'. In: *Journal of Machine Learning Research* 12.May, pp. 1771–1812.

Clifford, Peter (1990). 'Markov random fields in statistics'. In: *Disorder in physical systems: A volume in honour of John M. Hammersley* 19.

Combettes, Patrick L and Valérie R Wajs (2005). 'Signal recovery by proximal forward-backward splitting'. In: *Multiscale Modeling & Simulation* 4.4, pp. 1168–1200.

Damaraju, Eswar, Elena A Allen, Aysenil Belger, Judith M Ford, S McEwen, DH Mathalon, BA Mueller, GD Pearlson, SG Potkin, A Preda et al. (2014). 'Dynamic functional connectivity analysis reveals transient states of dysconnectivity in schizophrenia'. In: *NeuroImage: Clinical* 5, pp. 298–308.

Danaher, Patrick, Pei Wang and Daniela M Witten (2014). 'The joint graphical lasso for inverse covariance estimation across multiple classes'. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 76.2, pp. 373–397.

Defays, Daniel (1977). 'An efficient algorithm for a complete link method'. In: *The Computer Journal* 20.4, pp. 364–366.

Dempster, Arthur P (1972). 'Covariance selection'. In: *Biometrics*, pp. 157–175.

Dempster, Arthur P, Nan M Laird and Donald B Rubin (1977). 'Maximum likelihood from incomplete data via the EM algorithm'. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 39.1, pp. 1–22.

Elwert, Felix (2013). 'Graphical causal models'. In: *Handbook of causal analysis for social research*. Springer, pp. 245–273.

Epskamp, Sacha (2015). *IsingSampler: Sampling Methods and Distribution Functions for the Ising moel*. R package version 0.2. URL: `https://CRAN.R-project.org/package=IsingSampler`.

Epskamp, Sacha, Denny Borsboom and Eiko I Fried (2018). 'Estimating psychological networks and their accuracy: A tutorial paper'. In: *Behavior Research Methods* 50.1, pp. 195–212.

Erdős, Paul and Alfréd Rényi (1960). 'On the evolution of random graphs'. In: *Publ. Math. Inst. Hung. Acad. Sci* 5.1, pp. 17–60.

Fan, Roger, Byoungwook Jang, Yuekai Sun and Shuheng Zhou (2019). 'Precision Matrix Estimation with Noisy and Missing Data'. In: *arXiv preprint arXiv:1904.03548*.

Farasat, Alireza, Alexander Nikolaev, Sargur N Srihari and Rachael Hageman Blair (2015). 'Probabilistic graphical models in modern social network analysis'. In: *Social Network Analysis and Mining* 5.1, p. 62.

Foti, Nicholas J, Rahul Nadkarni, AK Lee and Emily B Fox (2016). 'Sparse plus low-rank graphical models of time series for functional connectivity in MEG'. In: *2nd KDD Workshop on Mining and Learning from Time Series*.

Fox, Emily B and Mike West (2011). 'Autoregressive models for variance matrices: Stationary inverse Wishart processes'. In: *arXiv preprint arXiv:1107.5239*.

Foygel, Rina and Mathias Drton (2010). 'Extended Bayesian information criteria for Gaussian graphical models'. In: *Advances in neural information processing systems*, pp. 604–612.

Frey, Brendan J (2002). 'Extending factor graphs so as to unify directed and undirected graphical models'. In: *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers Inc., pp. 257–264.

Friedman, Jerome, Trevor Hastie and Robert Tibshirani (2008). 'Sparse inverse covariance estimation with the graphical lasso'. In: *Biostatistics* 9.3, pp. 432–441.

Friedman, Nir (1998). 'The Bayesian structural EM algorithm'. In: *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., pp. 129–138.

Fuso, Andrea, Vincenzina Nicolia, Rosaria A Cavallaro and Sigfrido Scarpa (2011). 'DNA methylase and demethylase activities are modulated by one-carbon metabolism in Alzheimer's disease models'. In: *The Journal of nutritional biochemistry* 22.3, pp. 242–251.

Geman, Stuart and Donald Geman (1987). 'Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images'. In: *Readings in computer vision*. Elsevier, pp. 564–584.

Geng, Sinong, Zhaobin Kuang, Peggy Peissig and David Page (2018). 'Temporal Poisson Square Root Graphical Models'. In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. Stockholmsmässan, Stockholm Sweden: PMLR, pp. 1714–1723. URL: http://proceedings.mlr.press/v80/geng18a.html.

Gibberd, Alex J and Sandipan Roy (2017). 'Multiple changepoint estimation in high-dimensional gaussian graphical models'. In: *arXiv preprint arXiv:1712.05786*.

Greenshtein, Eitan, Ya'Acov Ritov et al. (2004). 'Persistence in high-dimensional linear predictor selection and the virtue of overparametrization'. In: *Bernoulli* 10.6, pp. 971–988.

Guo, Jian, Elizaveta Levina, George Michailidis and Ji Zhu (2011). 'Joint estimation of multiple graphical models'. In: *Biometrika* 98.1, pp. 1–15.

Hagberg, Aric, Pieter Swart and Daniel S Chult (2008). *Exploring network structure, dynamics, and function using NetworkX*. Tech. rep. Los Alamos National Lab.(LANL), Los Alamos, NM (United States).

Hallac, David, Jure Leskovec and Stephen Boyd (2015). 'Network lasso: Clustering and optimization in large graphs'. In: *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, pp. 387–396.

Hallac, David, Youngsuk Park, Stephen Boyd and Jure Leskovec (2017a). 'Network inference via the time-varying graphical lasso'. In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, pp. 205–213.

Hallac, David, Sagar Vare, Stephen Boyd and Jure Leskovec (2017b). 'Toeplitz inverse covariance-based clustering of multivariate time series data'. In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, pp. 215–223.

Hämmerle, Barbara, Yania Yañez, Sarai Palanca, Adela Cañete, Deborah J Burks, Victoria Castel and Jaime Font de Mora (2013). 'Targeting neuroblastoma stem cells with retinoic acid and proteasome inhibitor'. In: *PloS one* 8.10, e76761.

Harutyunyan, Hrayr, Daniel Moyer, Hrant Khachatrian, Greg Ver Steeg and Aram Galstyan (2019). 'Efficient Covariance Estimation from Temporal Data'. In: *arXiv preprint arXiv:1905.13276*.

Hastie, Trevor, Robert Tibshirani and Martin Wainwright (2015). *Statistical learning with sparsity: the lasso and generalizations*. Chapman and Hall/CRC.

Hecker, Michael, Sandro Lambeck, Susanne Toepfer, Eugene Van Someren and Reinhard Guthke (2009). 'Gene regulatory network inference: data integration in dynamic models—a review'. In: *Biosystems* 96.1, pp. 86–103.

Hertz, John, Yasser Roudi and Joanna Tyrcha (2011). 'Ising models for inferring network structure from spike data'. In: *arXiv preprint arXiv:1106.1752*.

Ho, Qirong, Le Song and Eric Xing (2011). 'Evolving cluster mixed-membership blockmodel for time-evolving networks'. In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 342–350.

Honorio, Jean and Dimitris Samaras (2010). 'Multi-Task Learning of Gaussian Graphical Models.' In: *ICML*. Citeseer, pp. 447–454.

Horn, Roger A and Charles R Johnson (2012). *Matrix analysis*. Cambridge university press.

Huang, Lei, Li Liao and Cathy H Wu (2016). 'Inference of protein-protein interaction networks from multiple heterogeneous data'. In: *EURASIP Journal on Bioinformatics and Systems Biology* 2016.1, p. 8.

Ising, Ernst (1925). 'Beitrag zur theorie des ferromagnetismus'. In: *Zeitschrift für Physik A Hadrons and Nuclei* 31.1, pp. 253–258.

Jalali, Ali, Pradeep Ravikumar, Vishvas Vasuki and Sujay Sanghavi (2011). 'On learning discrete graphical models using group-sparse regularization'. In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 378–387.

Jones, David T, Daniel WA Buchan, Domenico Cozzetto and Massimiliano Pontil (2011). 'PSICOV: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments'. In: *Bioinformatics* 28.2, pp. 184–190.

Jones, Eric, Travis Oliphant, Pearu Peterson et al. (2001–). *SciPy: Open source scientific tools for Python*. [Online; accessed]. URL: `http://www.scipy.org/`.

Kanehisa, Minoru and Susumu Goto (2000). 'KEGG: kyoto encyclopedia of genes and genomes'. In: *Nucleic acids research* 28.1, pp. 27–30.

Karlis, Dimitris (2003). 'An EM algorithm for multivariate Poisson distribution and related models'. In: *Journal of Applied Statistics* 30.1, pp. 63–77.

Kolar, Mladen, Le Song, Amr Ahmed, Eric P Xing et al. (2010). 'Estimating time-varying networks'. In: *The Annals of Applied Statistics* 4.1, pp. 94–123.

Kotz, Samuel, Tomasz Kozubowski and Krzystof Podgorski (2012). *The Laplace distribution and generalizations: a revisit with applications to communications, economics, engineering, and finance*. Springer Science & Business Media.

Lam, Clifford and Jianqing Fan (2009). 'Sparsistency and rates of convergence in large covariance matrix estimation'. In: *Annals of statistics* 37.6B, p. 4254.

Lange, Tilman, Volker Roth, Mikio L Braun and Joachim M Buhmann (2004). 'Stability-based validation of clustering solutions'. In: *Neural computation* 16.6, pp. 1299–1323.

Lauritzen, Steffen L (1996). *Graphical models*. Vol. 17. Clarendon Press.

Lee, Jason D and Trevor J Hastie (2015). 'Learning the structure of mixed graphical models'. In: *Journal of Computational and Graphical Statistics* 24.1, pp. 230–253.

Li, Xiao, Jinzhu Jia and Yuan Yao (2015). 'Mixed and missing data: a unified treatment with latent graphical models'. In: *arXiv preprint arXiv:1511.04656*.

Li, Zehang Richard, Tyler H McCormick and Samuel J Clark (2018). 'Bayesian joint spike-and-slab graphical lasso'. In: *arXiv preprint arXiv:1805.07051*.

Lipska, Beata S, Elżbieta Drożynska, Paola Scaruffi, Gian Paolo Tonini, Ewa Iżycka-Świeszewska, Szymon Ziętkiewicz, Anna Balcerska, Danuta Perek, Alicja Chybicka, Wojciech Biernat et al. (2009). 'c. 1810C> T Polymorphism of NTRK1 Gene is associated with reduced Survival in Neuroblastoma Patients'. In: *BMC cancer* 9.1, p. 436.

Little, Roderick JA and Donald B Rubin (2019). *Statistical analysis with missing data*. Vol. 793. Wiley.

Liu, Han, Fang Han and Cun-hui Zhang (2012). 'Transelliptical graphical models'. In: *Advances in neural information processing systems*, pp. 800–808.

Liu, Han, John Lafferty and Larry Wasserman (2009). 'The nonparanormal: Semiparametric estimation of high dimensional undirected graphs'. In: *Journal of Machine Learning Research* 10.Oct, pp. 2295–2328.

Liu, Han, Kathryn Roeder and Larry Wasserman (2010). 'Stability approach to regularization selection (stars) for high dimensional graphical models'. In: *Advances in neural information processing systems*, pp. 1432–1440.

Liu, Han, Fang Han, Ming Yuan, John Lafferty, Larry Wasserman et al. (2012). 'High-dimensional semiparametric Gaussian copula graphical models'. In: *The Annals of Statistics* 40.4, pp. 2293–2326.

Ma, Shiqian, Lingzhou Xue and Hui Zou (2013). 'Alternating direction methods for latent variable Gaussian graphical model selection'. In: *Neural computation* 25.8, pp. 2172–2198.

Madow, William G, Harold Nisselson and Ingram Olkin (1983). 'Incomplete data in sample surveys. Vol. 1: Report and case studies'. In:

Marsh, Eric D, Bradley Peltzer, Merritt W Brown III, Courtney Wusthoff, Phillip B Storm Jr, Brian Litt and Brenda E Porter (2010). 'Interictal EEG spikes identify the region of electrographic seizure onset in some, but not all, pediatric epilepsy patients'. In: *Epilepsia* 51.4, pp. 592–601.

Meinshausen, Nicolai and Peter Bühlmann (2010). 'Stability selection'. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 72.4, pp. 417–473.

Meinshausen, Nicolai, Peter Bühlmann et al. (2006). 'High-dimensional graphs and variable selection with the lasso'. In: *The annals of statistics* 34.3, pp. 1436–1462.

Meng, Zhaoshi, Brian Eriksson and Al Hero (2014). 'Learning latent variable Gaussian graphical models'. In: *International Conference on Machine Learning*, pp. 1269–1277.

Metzker, Michael L (2010). 'Sequencing technologies—the next generation'. In: *Nature reviews genetics* 11.1, p. 31.

Milenković, Tijana and Nataša Pržulj (2008). 'Uncovering biological network function via graphlet degree signatures'. In: *Cancer informatics* 6, CIN–S680.

Moghaddam, Baback, Emtiyaz Khan, Kevin P Murphy and Benjamin M Marlin (2009). 'Accelerating Bayesian structural inference for non-decomposable Gaussian graphical models'. In: *Advances in Neural Information Processing Systems*, pp. 1285–1293.

Mohan, Karthik, Mike Chung, Seungyeop Han, Daniela Witten, Su-In Lee and Maryam Fazel (2012). 'Structured learning of Gaussian graphical models'. In: *Advances in neural information processing systems*, pp. 620–628.

Molinaro, Annette M, Richard Simon and Ruth M Pfeiffer (2005). 'Prediction error estimation: a comparison of resampling methods'. In: *Bioinformatics* 21.15, pp. 3301–3307.

Molinelli, Evan J, Anil Korkut, Weiqing Wang, Martin L Miller, Nicholas P Gauthier, Xiaohong Jing, Poorvi Kaushik, Qin He, Gordon Mills, David B Solit et al. (2013). 'Perturbation biology: inferring signaling networks in cellular systems'. In: *PLoS computational biology* 9.12, e1003290.

Moulik, Nirmalya Roy, Archana Kumar and Suraksha Agrawal (2017). 'Folic acid, one-carbon metabolism & childhood cancer'. In: *The Indian journal of medical research* 146.2, p. 163.

Mrowczynski, Oliver D, AB Madhankumar, Becky Slagle-Webb, Sang Y Lee, Brad E Zacharia and James R Connor (2017). 'HFE genotype affects exosome phenotype in cancer'. In: *Biochimica et Biophysica Acta (BBA)-General Subjects* 1861.8, pp. 1921–1928.

Müller, Christian L, Richard Bonneau and Zachary Kurtz (2016). 'Generalized stability approach for regularized graphical models'. In: *arXiv preprint arXiv:1605.07072*.

Murphy, Kevin P (2012). *Machine learning: a probabilistic perspective*. MIT press.

Murphy, Kevin Patrick and Stuart Russell (2002). 'Dynamic bayesian networks: representation, inference and learning'. In:

Nelder, John Ashworth and Robert WM Wedderburn (1972). 'Generalized linear models'. In: *Journal of the Royal Statistical Society: Series A (General)* 135.3, pp. 370–384.

Nikoloulopoulos, Aristidis K and Dimitris Karlis (2009a). 'Finite normal mixture copulas for multivariate discrete data modeling'. In: *Journal of Statistical Planning and Inference* 139.11, pp. 3878–3890.

— (2009b). 'Modeling multivariate count data using copulas'. In: *Communications in Statistics-Simulation and Computation* 39.1, pp. 172–187.

Nussbaum, Frank and Joachim Giesen (2019). *Ising Models with Latent Conditional Gaussian Variables*. arXiv: `1901.09712 [cs.LG]`.

Oliphant, Travis (2006–). *NumPy: A guide to NumPy*. USA: Trelgol Publishing. [Online; accessed <today>]. URL: `http://www.numpy.org/`.

Orchard, Peter, Felix Agakov and Amos Storkey (2013). 'Bayesian inference in sparse Gaussian graphical models'. In: *arXiv preprint arXiv:1309.7311*.

Pedregosa, F. et al. (2011). 'Scikit-learn: Machine Learning in Python'. In: *Journal of Machine Learning Research* 12, pp. 2825–2830.

Pereira, José, Morteza Ibrahimi and Andrea Montanari (2010). 'Learning networks of stochastic differential equations'. In: *Advances in Neural Information Processing Systems*, pp. 172–180.

Politis, Dimitris N, Joseph P Romano and Michael Wolf (1999). *Subsampling*. Springer Science & Business Media.

Pržulj, Nataša (2007). 'Biological network comparison using graphlet degree distribution'. In: *Bioinformatics* 23.2, e177–e183.

Pržulj, Natasa, Derek G Corneil and Igor Jurisica (2004). 'Modeling interactome: scale-free or geometric?' In: *Bioinformatics* 20.18, pp. 3508–3515.

Rasmussen, Carl Edward (2003). 'Gaussian processes in machine learning'. In: *Summer School on Machine Learning*. Springer, pp. 63–71.

Ravikumar, Pradeep K, Garvesh Raskutti, Martin J Wainwright and Bin Yu (2009a). 'Model Selection in Gaussian Graphical Models: High-Dimensional Consistency of ℓ_1-regularized MLE'. In: *Advances in Neural Information Processing Systems*, pp. 1329–1336.

Ravikumar, Pradeep, Martin J Wainwright, John D Lafferty et al. (2010). 'High-dimensional Ising model selection using ℓ_1-regularized logistic regression'. In: *The Annals of Statistics* 38.3, pp. 1287–1319.

Ravikumar, Pradeep, John Lafferty, Han Liu and Larry Wasserman (2009b). 'Sparse additive models'. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 71.5, pp. 1009–1030.

Ravikumar, Pradeep, Martin J Wainwright, Garvesh Raskutti, Bin Yu et al. (2011). 'High-dimensional covariance estimation by minimizing ℓ_1-penalized log-determinant divergence'. In: *Electronic Journal of Statistics* 5, pp. 935–980.

Robin, Genevieve, Christophe Ambroise and Stéphane Robin (2018). 'Incomplete graphical model inference via latent tree aggregation'. In: *Statistical Modelling*, p. 1471082X18786289.

Rosenberg, Andrew and Julia Hirschberg (2007). 'V-measure: A conditional entropy-based external cluster evaluation measure'. In: *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, pp. 410–420.

Rothman, Adam J, Peter J Bickel, Elizaveta Levina, Ji Zhu et al. (2008). 'Sparse permutation invariant covariance estimation'. In: *Electronic Journal of Statistics* 2, pp. 494–515.

Roudi, Yasser, Joanna Tyrcha and John Hertz (2009). 'Ising model for neural data: model quality and approximate methods for extracting functional connectivity'. In: *Physical Review E* 79.5, p. 051915.

Sakamoto, Yosiyuki, Makio Ishiguro and Genshiro Kitagawa (1986). 'Akaike information criterion statistics'. In: *Dordrecht, The Netherlands: D. Reidel* 81.

Salzo, Saverio (2017). 'The variable metric forward-backward splitting algorithm under mild differentiability assumptions'. In: *SIAM Journal on Optimization* 27.4, pp. 2153–2181.

Sarajlić, Anida, Noël Malod-Dognin, Ömer Nebil Yaveroğlu and Nataša Pržulj (2016). 'Graphlet-based characterization of directed networks'. In: *Scientific reports* 6, p. 35098.

Schafer, Joseph L (1997). *Analysis of incomplete multivariate data*. Chapman and Hall/CRC.

Schneidman, Elad, Michael J Berry II, Ronen Segev and William Bialek (2006). 'Weak pairwise correlations imply strongly correlated network states in a neural population'. In: *Nature* 440.7087, p. 1007.

Schwing, Alexander, Tamir Hazan, Marc Pollefeys and Raquel Urtasun (2012). 'Efficient structured prediction with latent variables for general graphical models'. In: *arXiv preprint arXiv:1206.6436*.

Siegmund, David (2004). 'Model selection in irregular problems: Applications to mapping quantitative trait loci'. In: *Biometrika* 91.4, pp. 785–800.

Smith, Stephen M, Karla L Miller, Gholamreza Salimi-Khorshidi, Matthew Webster, Christian F Beckmann, Thomas E Nichols, Joseph D Ramsey and Mark W Woolrich (2011). 'Network modelling methods for FMRI'. In: *Neuroimage* 54.2, pp. 875–891.

Snoek, Jasper, Hugo Larochelle and Ryan P Adams (2012). 'Practical bayesian optimization of machine learning algorithms'. In: *Advances in neural information processing systems*, pp. 2951–2959.

Städler, Nicolas and Peter Bühlmann (2012). 'Missing values: sparse inverse covariance estimation and an extension to sparse regression'. In: *Statistics and Computing* 22.1, pp. 219–235.

Stegle, Oliver, Sarah A Teichmann and John C Marioni (2015). 'Computational and analytical challenges in single-cell transcriptomics'. In: *Nature Reviews Genetics* 16.3, p. 133.

Stoica, Petre and Yngve Selen (2004). 'Model-order selection: a review of information criterion rules'. In: *IEEE Signal Processing Magazine* 21.4, pp. 36–47.

Suebsoonthron, Junjira, Thiranut Jaroonwitchawan, Montarop Yamabhai and Parinya Noisa (2017). 'Inhibition of WNT signaling reduces differentiation and induces sensitivity to doxorubicin in human malignant neuroblastoma SH-SY5Y cells'. In: *Anti-cancer drugs* 28.5, pp. 469–479.

Tibshirani, Robert (1996). 'Regression shrinkage and selection via the lasso'. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 58.1, pp. 267–288.

Tomasi, Federico, Veronica Tozzo, Alessandro Verri and Saverio Salzo (2018a). 'Forward-Backward Splitting for Time-Varying Graphical Models'. In: *International Conference on Probabilistic Graphical Models*, pp. 475–486.

Tomasi, Federico, Veronica Tozzo, Saverio Salzo and Alessandro Verri (2018b). 'Latent variable time-varying network inference'. In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, pp. 2338–2346.

Totaro, Francesca, Flora Cimmino, Piero Pignataro, Giovanni Acierno, Marilena De Mariano, Luca Longo, Gian Paolo Tonini, Achille Iolascon and Mario Capasso (2013). 'Impact of interleukin-6–174 G> C gene promoter polymorphism on neuroblastoma'. In: *PloS one* 8.10, e76810.

Tozzo, Veronica, Federico Tomasi, Margherita Squillario and Annalisa Barla (2018). 'Group induced graphical lasso allows for discovery of molecular pathways-pathways interactions'. In: *Proceedings of Machine Learning 4 Health NeurIPS 2018*.

Tran, Nguyen Q and Alexander Jung (2018). 'On the sample complexity of graphical model selection from non-stationary samples'. In: *2018 IEEE In-*

ternational Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, pp. 6314–6317.

Varoquaux, Gaël, Alexandre Gramfort, Jean-Baptiste Poline and Bertrand Thirion (2010). 'Brain covariance selection: better individual functional connectivity models using population prior'. In: *Advances in neural information processing systems*, pp. 2334–2342.

Vinci, Giuseppe, Valérie Ventura, Matthew A Smith, Robert E Kass et al. (2018). 'Adjusted regularization in latent graphical models: Application to multiple-neuron spike count data'. In: *The Annals of Applied Statistics* 12.2, pp. 1068–1095.

Von Luxburg, Ulrike et al. (2010). 'Clustering stability: an overview'. In: *Foundations and Trends® in Machine Learning* 2.3, pp. 235–274.

Wainwright, Martin J, Michael I Jordan et al. (2008). 'Graphical models, exponential families, and variational inference'. In: *Foundations and Trends® in Machine Learning* 1.1–2, pp. 1–305.

Wainwright, Martin J, John D Lafferty and Pradeep K Ravikumar (2007). 'High-Dimensional Graphical Model Selection Using ℓ_1-Regularized Logistic Regression'. In: *Advances in neural information processing systems*, pp. 1465–1472.

Wan, Ying-Wooi, Genevera I Allen, Yulia Baker, Eunho Yang, Pradeep Ravikumar, Matthew Anderson and Zhandong Liu (2016). 'XMRF: an R package to fit Markov Networks to high-throughput genetics data'. In: *BMC systems biology* 10.3, p. 69.

Wang, Jing, Dexter Duncan, Zhiao Shi and Bing Zhang (2013). 'WEB-based GEne SeT AnaLysis Toolkit (WebGestalt): update 2013'. In: *Nucleic Acids Research* 41.W1, W77–W83. DOI: 10.1093/nar/gkt439. eprint: /oup/backfile/content_public/journal/nar/41/w1/10.1093_nar_gkt439/3/gkt439.pdf. URL: http://dx.doi.org/10.1093/nar/gkt439.

Wang, Xiqian, Jing Li, Xiao Xu, Jiachun Zheng and Qingbo Li (2018). 'miR-129 inhibits tumor growth and potentiates chemosensitivity of neuroblastoma by targeting MYO10'. In: *Biomedicine & Pharmacotherapy* 103, pp. 1312–1318.

Wasserman, Larry and Kathryn Roeder (2009). 'High dimensional variable selection'. In: *Annals of statistics* 37.5A, p. 2178.

Weinberger, Kilian Q, Fei Sha, Qihui Zhu and Lawrence K Saul (2007). 'Graph Laplacian regularization for large-scale semidefinite programming'. In: *Advances in neural information processing systems*, pp. 1489–1496.

Wilson, Andrew Gordon and Zoubin Ghahramani (2011). 'Generalised Wishart processes'. In: *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*. AUAI Press, pp. 736–744.

Xie, Yuying, Yufeng Liu and William Valdar (2016). 'Joint estimation of multiple dependent Gaussian graphical models with applications to mouse genomics'. In: *Biometrika* 103.3, pp. 493–511.

Yang, Eunho, Genevera Allen, Zhandong Liu and Pradeep K. Ravikumar (2012). 'Graphical Models via Generalized Linear Models'. In: *Advances in Neural Information Processing Systems 25*. Ed. by F. Pereira, C. J. C. Burges, L. Bottou and K. Q. Weinberger. Curran Associates, Inc., pp. 1358–1366.

URL: http://papers.nips.cc/paper/4617-graphical-models-via-generalized-linear-models.pdf.

Yang, Eunho, Pradeep K Ravikumar, Genevera I Allen and Zhandong Liu (2013). 'On Poisson graphical models'. In: *Advances in Neural Information Processing Systems*, pp. 1718–1726.

Yang, Eunho, Yulia Baker, Pradeep Ravikumar, Genevera Allen and Zhandong Liu (2014). 'Mixed graphical models via exponential families'. In: *Artificial Intelligence and Statistics*, pp. 1042–1050.

Yang, Eunho, Pradeep Ravikumar, Genevera I Allen and Zhandong Liu (2015). 'Graphical models via univariate exponential family distributions'. In: *The Journal of Machine Learning Research* 16.1, pp. 3813–3847.

Yu, W., M. Clyne, M. J. Khoury and M. Gwinn (2010). 'Phenopedia and Genopedia: disease-centered and gene-centered views of the evolving knowledge of human genetic associations'. In: *Bioinformatics* 26.1, pp. 145–146. DOI: 10.1093/bioinformatics/btp618. eprint: /oup/backfile/content_public/journal/bioinformatics/26/1/10.1093_bioinformatics_btp618/2/btp618.pdf. URL: http://dx.doi.org/10.1093/bioinformatics/btp618.

Yuan, Ming (2012). 'Discussion: Latent variable graphical model selection via convex optimization'. In: *The Annals of Statistics* 40.4, pp. 1968–1972.

Yuan, Ming and Yi Lin (2007). 'Model selection and estimation in the Gaussian graphical model'. In: *Biometrika* 94.1, pp. 19–35.

Zhao, Qian, Mei Jin, Da-Wei Zhang, Wen Zhao, Xi-Si Wang, Zhi-Xia Yue, Chao Duan, Cheng Huang and Xiao-Li Ma (2018). 'Serum Interleukin-6 Level and the rs1800795 Polymorphism in its Gene Associated with Neuroblastoma Risk in Chinese Children'. In: *Chinese medical journal* 131.9, p. 1075.

Žitnik, Marinka and Blaž Zupan (2015). 'Gene network inference by fusing data from diverse distributions'. In: *Bioinformatics* 31.12, pp. i230–i239.

Zou, Hui, Trevor Hastie and Robert Tibshirani (Oct. 2007). 'On the "degrees of freedom" of the lasso'. In: *Ann. Statist.* 35.5, pp. 2173–2192. DOI: 10.1214/009053607000000127. URL: https://doi.org/10.1214/009053607000000127.

Acronyms

ADMM - Alternating Direction Method of Multipliers

BA - Balanced Accuracy

FBS - Forward Backward Splitting

GL - Graphical Lasso

GGM - Gaussian Graphical Model

EM - Expectation Maximisation

IGM - Ising Graphical Model

JGL - Joint Graphical Lasso

KEGG - Kyoto Encyclopedia of Genes and Genomes

LGL - Latent Graphical Lasso

LTGL_{κ} - Latent Temporal Graphical Lasso

LVGLASSO - Latent Variable Graphical Lasso

MAR - Missing At Random

MissGL - Missing Graphical Lasso

MLE - Maximum Likelihood Estimation

MRF - Markov Random Field

MTGL_{G} - Missing Temporal Graphical Lasso with Group prior

MTGL_{κ} - Kernel Missing Temporal Graphical Lasso

MTGL^{L} - Missing Temporal Graphical Lasso with Latent data

MTGL^{P} - Missing Temporal Graphical Lasso with Partial data

REGAIN - REgularised GrAph INference library

PGM - Poisson Graphical Model

TIGM$_\kappa$ - Kernel Temporal Ising Model

TIGM$_P$ - Temporal Ising Model with automatic Pattern identification

TGL - Temporal Graphical Lasso

TGL$_\kappa$ - Kernel Temporal Graphical Lasso

TGL$_P$ - Temporal Graphical Lasso with automatic Pattern identification

TPGM$_\kappa$ - Kernel Temporal Poisson Model

TPGM$_P$ - Temporal Poisson Model with automatic Pattern identification

TCGA - The Cancer Genome Atlas

TVGL - Time-Varying Graphical Lasso

WP - Wishart Process

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements.

Genova, Italy
December 2019

Veronica Tozzo