A Bayesian Latent Time-Series Model for Switching Temporal Interaction Analysis

by

Zoran Dzunic

B.S., Electrical Engineering and Computer Science, University of Nis, 2005
S.M., Electrical Engineering and Computer Science, MIT, 2009

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Electrical Engineering and Computer Science
at the Massachusetts Institute of Technology

February 2016

Signature of Author: Signature redacted

Department of Electrical Engineering and Computer Science
January 29, 2016

Certified by: Signature redacted

John W. Fisher III
Senior Research Scientist, Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by: Signature redacted

Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Committee for Graduate Students

A Bayesian Latent Time-Series Model for Switching Temporal Interaction Analysis

by Zoran Dzunic

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

Abstract

We introduce a Bayesian discrete-time framework for switching-interaction analysis under uncertainty, in which latent interactions, switching pattern and signal states and dynamics are inferred from noisy and possibly missing observations of these signals. We propose reasoning over posterior distribution of these latent variables as a means of combating and characterizing uncertainty. This approach also allows for answering a variety of questions probabilistically, which is suitable for exploratory pattern discovery and post-analysis by human experts. This framework is based on a Bayesian learning of the structure of a switching dynamic Bayesian network (DBN) and utilizes a state-space approach to allow for noisy observations and missing data. It generalizes the autoregressive switching interaction model of Siracusa et al. [50], which does not allow observation noise, and the switching linear dynamic system model of Fox et al. [16], which does not infer interactions among signals.

We develop a Gibbs sampling inference procedure, which is particularly efficient in the case of linear Gaussian dynamics and observation models. We use a modular prior over structures and a bound on the number of parent sets per signal to reduce the number of structures to consider from super-exponential to polynomial. We provide a procedure for setting the parameters of the prior and initializing latent variables that leads to a successful application of the inference algorithm in practice, and leaves only few general parameters to be set by the user. A detailed analysis of the computational and memory complexity of each step of the algorithm is also provided.

We demonstrate the utility of our framework on different types of data. Different benefits of the proposed approach are illustrated using synthetic data. Most real data do not contain annotation of interactions. To demonstrate the ability of the algorithm to infer interactions and the switching pattern from time-series data in a realistic setting, joystick data is created, which is a controlled, human-generated data that implies ground truth annotations by design. Climate data is a real data used to illustrate the variety of applications and types of analyses enabled by the developed methodology.

Finally, we apply the developed model to the problem of structural health monitoring in civil engineering. Time-series data from accelerometers located at multiple positions on a building are obtained for two laboratory model structures and a real building. We analyze the results of interaction analysis and how the inferred dependen-

cies among sensor signals relate to the physical structure and properties of the building, as well as the environment and excitation conditions. We develop time-series classification and single-class classification extensions of the model and apply them to the problem of damage detection. We show that the method distinguishes time-series obtained under different conditions with high accuracy, in both supervised and single-class classification setups.

Thesis Supervisor: John W. Fisher III
Title: Senior Research Scientist, Electrical Engineering and Computer Science

Acknowledgments

I would like to thank my advisor, John Fisher, for providing me guidance and support at every step of this road. His ideas and enthusiasm have been instrumental not only for my work, but also for broadening my views and growing as a researcher. I enjoyed numerous conversations with him over the years and timely jokes that he would often insert. I would also like to thank Bill Freeman and Asu Ozdaglar, members of my thesis committee, for questioning my work and providing me with invaluable comments that vastly improved the text of this thesis.

This thesis would not have been possible without help from other researchers. I would like to personally thank Justin Chen and Professor Oral Buyukozturk from MIT Civil Engineering Department, as well as Hossein Mobahi from the SLI group, with whom I had a fruitful collaboration on the structural health monitoring project. I owe big thank to Michael Siracusa, whose work I continued. He was so kind to meet with me many times to discuss his work, his code, and possibilities for then future work, which tremendously helped me get started on my own project. I also greatly enjoyed mentoring Bonny Jain, who worked on an extension of my model. I learned a lot from that experience.

Other members of the SLI group have always been there to help me. and they became really good friends. In no particular order, I thank Dahua Lin, Giorgos Papachristoudis, Randi Cabezas, Sue Zheng, Julian Straub, Christopher Dean, Oren Freifeld, Guy Rosman, David Hayden, Vadim Smolyakov, and Aryan Khojandi.

My everyday life at MIT has been joy thanks to my officemates. I would like to thank Ramesh Sridharan, George Chen, Giorgos Papachristoudis, Adrian Dalca, Danielle Pace, Polina Binder, Guy Rosman and Danial Lashkari for always being ready to help, talk and have fun, and for being the best officemates I could have asked for.

Finally, I would like to thank all of my friends and family. I especially thank my parents for their endless support. Most of all, I thank my lovely wife Ivana and my lovely daughter Lenka for their love, support and patience and for giving me a reason to look towards the future.

Different aspects of this thesis were partially supported by the Office of Naval Research Multidisciplinary Research Initiative program award N000141110688, the Army Research Office Multidisciplinary Research Initiative program award W911NF-11-1-

0391, and Shell via the MIT Energy Initiative.

Contents

List of Figures

Chapter 1

Introduction

EXAMPLES of interaction can be found everywhere. One can talk about an interaction of people in a social network, at an event, or in a street, interaction of companies on a stock market, neurons in a brain, climate indices, and so on. Learning such interactions is important, as that can further our understanding of the processes among the involved entities, as well as lead to novel applications. However, while some interactions can be easily detected by our senses, a lot of them are still hard to identify by humans. Therefore, different sciences focus on inferring and analyzing interactions of different types and in different domains from data that can be related to interactions.

In this thesis, we consider the problem of inference over interactions from time-series data. The notion of interaction may be defined differently in different disciplines. For example, interaction between two objects often assumes a two-way influence between them. When more than two objects are involved, this would imply a two-way influence between any pair of objects, and inferring interactions would reduce to inferring groups (cliques) of objects that interact among each other. We are, however, interested in a more general case, in which an interaction is defined as any set of directed (one-way) influences among objects and the goal is to uncover such set of relationships, which we refer to as the **structure of interaction**. More formally, an **interaction graph** is defined as a directed graph $G = (V, E)$, where V is the set of nodes that correspond to objects, and E is the set of edges that correspond to directed influences [50]. In other words, $i \to j \in E$ if object i *influences* (has an effect on) object j, in which case we also say that object j *depends* on object i. We refer to the set of edges of the interaction graph, E, as the interaction structure. In addition, we make the following assumptions:

- Dependencies that constitute an interaction are **temporal causal relationships** [44], meaning that the behavior an object can only influence the future behavior of another object (or set of objects).
- Objects are represented as **multivariate time-series** (discrete-time multivariate signals). Therefore, we will often talk more abstractly about the interaction among time-series, or signals, where it will be assumed that these signals correspond to some objects or abstract entities, whose interaction is a subject of interest.[1]

[1]Note that we have not done analysis on the relationship between object representation (in terms

Learning temporal interactions from time-series data is challenging for several reasons:

- The number of possible interactions among a set of signals is extremely large – super-exponential in the number of signals. Namely, if N is the number of signals, the number of possible interactions among them is equal to the number of different directed graphs, which is 2^{N^2}.
- Interactions may change over time, and therefore the problem of learning interaction becomes the problem of learning different interactions at different points in time and the pattern of switching between these interactions.
- Underlying time-series are often not observed directly, but rather through some noisy observation process. In addition, data is sometimes missing due to an error or inability to collect observations at certain time points.

The first two problems have been addressed by the work of Siracusa and Fisher [49, 50], in which they develop a Bayesian switching temporal interaction model for inference over dynamically-varying temporal interaction structure from time-series data. However, their model assumes that time-series are observed directly and does not address the problem of noisy observations. On the other hand, switching state-space models have been used to learn switching joint dynamics of time-series from noisy data (e.g., [16, 22]), but these models do not learn interactions among time-series. Our goal is to fill in the gap and develop a method that addresses all three challenges above in a single framework. To that end, we develop a **state-space switching interaction model (SSIM)** [13], which combines the two approaches, as well as an efficient Gibbs sampling algorithm for inference over latent time-series, interactions and the switching pattern from noisy and (possibly) missing data in this model.

1.1 Bayesian Approach

In addition to the assumptions above, we also assume that there exists a discrete-time stochastic process that generates future observations of time-series from their past observations, such that each time-series possibly depends only on a subset of other time-series. This naturally leads to a **dynamic Bayesian network (DBN) representation** of the joint time-series model, and the problem of inference over switching interaction is reduced to the problem of inference over a switching DBN structure (as in [50]), which is depicted in Figure 1.1. A fist-order model, in which the dependency is only on the values at the previous time point, is illustrated for simplicity. Moreover, we will first derive the SSIM model with a first-order dependency among time-series. However, we will later extend the model to allow higher-order dependencies.

of feature representation and sampling frequency) and the ability to infer temporal interactions using statistical methods.

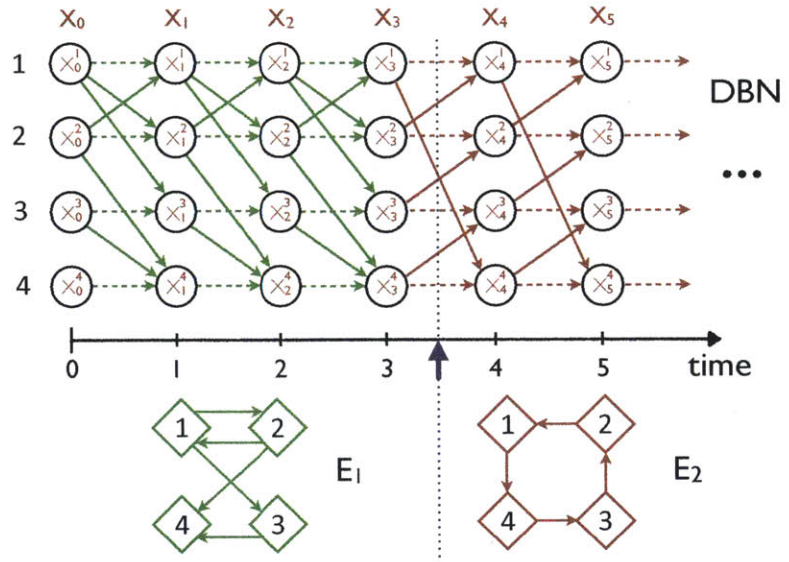


Figure 1.1: Dynamic Bayesian Network (DBN) representation of switching interaction among four signals. They initially evolve according to interaction graph E_1. At time point 4, the interaction pattern changes, and they evolve according to interaction graph E_2. Self-edges are assumed.

Inferring the structure of a (static or dynamic) Bayesian network presents a formidable challenge owing to the super-exponential number of possible directed graphs. It is known that the exact inference over such structures is NP-hard in general [10]. A number of heuristic methods for finding a structure with the maximum a posteriori (MAP) probability have been developed [7, 11, 27]. However, MAP estimates of network structures are known to be brittle. With limited data available, there may exist a large number of structures that explain the data well. Point estimates of structure (e.g., MAP) are likely to yield incorrect interactions. The problem is exacerbated when the structure varies over time and time-series state is not observed directly, but rather by some noisy observation process. To alleviate this, sampling approaches have been typically used to approximate the posterior distribution over structures with a number of samples from that distribution [36]. Due to the a typically highly-multimodal posterior landscape, efforts have been made to develop robust sampling algorithms that do not get stuck in local optima [19, 25, 39]. On the other hand, Siracusa and Fisher [50] use a modular prior assumption, which effectively allows independent inference over parent sets of each time-series (that can be done in exponential time), and additional constraints on possible parent sets (e.g., bounded in-degree), which result in a polynomial-time exact inference over a non-switching dynamic Bayesian structure, thus avoiding sampling over structure. These assumptions have also been exploited in the context of static Bayesian networks, but since such networks must be acyclic, a topological order of nodes must either be known a priori [7, 11, 27] or sampled [19].

We adopt the approach of Siracusa and Fisher [50] and use a modular bounded-indegree prior on the interaction structure, which allows for efficient inference over structures. Also, as discussed above, computing a posterior distribution over structures quantifies the uncertainty in structure, and also allows for a robust estimate of structural events, which are often of primary interest. Examples of such events are: "Does object A depend on object B, given that it interacts with object C?" or "Which object is the most influential, i.e., has the most objects that depend on it?".

Note that the exact inference over structures is possible only if there is no switching and observations are assumed perfect. In case of switching and/or observation noise, exact joint inference over latent time-series, switching pattern and interaction structures is intractable. Similarly as in [50], we use a Gibbs sampling approach to joint inference over these variables, in which an exact inference over interaction structures is performed when conditioned on other variables. However, unlike [50], where switching patterns obtained from different samples are aligned to produce a single most likely switching pattern, we reason over the distribution of switching patterns. This allows for computing statistics over switching patterns, such as the probability of two time points being in the same switching state. Consequently, there is no posterior distribution over structures defined for each switching state, since switching states are not aligned across samples. Instead, the switching pattern is marginalized out, and the posterior distribution over structure is computed for each time point separately, as it can indeed be different at each time point as a result of marginalization.

1.2 Contributions

The main contribution of the thesis is the introduction of a new model, which we refer to as the switching state-space interaction model (SSIM), and development of an efficient algorithm for Bayesian inference over switching interaction structure among time-series from noisy and possibly missing data [13], whereas the previous work assumes perfect observations [49, 50]. There are many examples where time-series measurements are noisy, such most data obtained through sensing, that motivate our method. For example, tracking objects in a video necessarily introduces observation noise, regardless of whether it is done by a human or an automatic tracker. Also, observations sometimes cannot be made due to occlusions, which results in missing data.

We introduce a linear-Gaussian variant of the SSIM, in which both time-series dependence and observation models are assumed linear and Gaussian. This specialization of the model is widely applicable and enables a particularly efficient inference procedure. We also introduce a latent-autoregressive linear-Gaussian SSIM, in which dependencies on an arbitrary number of previous time points are allowed. This extension is critical for many practical applications as first order models are often not sufficient to capture important dependencies. These two variants can be paralleled to analogous variants of the model of Siracusa and Fisher [49, 50], with the main distinction that their model does not incorporate an observation model.

Our approach extents the method of Siracusa and Fisher [49, 50] by introducing an observation model and assuming that the underlying time-series are in the latent space. While this extension is conceptually simple and intuitive, it poses several challenges that we address in this thesis. First, an additional step in an inference procedure for sampling latent time-series must be taken. Sequential sampling of state sequences is know to converge slowly. Batch sampling can be done efficiently using an exact message-passing algorithm only for some choices of dependence and observation models. For example, this is the case when linear-Gaussian models are used. Otherwise, approximate methods, such as particle filtering [2], must be employed. We take the advantage of the linear-Gaussian model and employ it in our work for efficient inference. However, a standard message-passing algorithm for sampling latent time-series shows to be numerically unstable in cases when data is missing, in particular when there are several consecutive time-points for which data is missing. To alleviate that, we develop an alternative message-passing algorithm for this step that uses a different representation and computation of messages that is numerically stable. Second, the latent space in the SSIM model is very complex – latent interactions, switching pattern and time-series, as well as parameters of dependence and observation models need to be inferred from noisy and possibly missing observations. Jointly, these variables create a complex probability space. The posterior distribution over these variables is highly multimodal and there could be different suboptimal explanations of the data. For example, high variance of the dependence or the observation model can explain the data well, but that is not the explanation that is typically sought. Also, assigning time points to switching states is effectively a clustering problem, and spaces of clusterings typically have multiple local optima. To avoid undesired local optima and steer the inference into the regions of posterior distribution that are of interest, we develop specific methods for setting the prior and initializing latent variables. In addition, we often use multiple restarts to improve the coverage of the posterior distributions with samples. These methods lead to an algorithm that is mostly free of tuning, except for a few general model parameters. The new way of setting the prior also improves the previous method of Siracusa and Fisher [49, 50].

We demonstrate the utility of our approach on several datasets. Synthetic data is generated to emphasize the advantage over other methods. Specifically, we show that inference over the interaction structure as a graph is necessary, and that simply analyzing pairwise dependencies separately (as in Granger causality tests [24]) may lead to a detection of spurious dependencies. We also show that our approach is advantageous over the previous method that does not account for observation noise [50] on an example in which the previous method assigns high probability to a spurious parent of a signal, because the correct parent does not predict well that signal alone due to the observation noise. When the observation noise is accounted for (our approach), the probability of a spurious edge is significantly reduced.

Unfortunately, real datasets typically do not contain ground truth interactions. Interactions are not know and are also difficult to annotate by humans due to their

complexity. This is, in the first place, a reason why learning interactions from data is an important task. However, it also renders testing inferred interactions difficult. To be able to test the results of interaction inference on non-synthetic data, we develop a new dataset, called joystick data, in which interactions and a switching pattern are known by design. Namely, five human players control points on a screen via joystick according to predefined tasks and a switching pattern between tasks. For example, a player can have an assignment to "follow" another player or to stay in the middle of the line between two other players. Therefore, interactions are implied by the tasks. We show that our method assigns high probability to the correct interactions and a switching pattern, and assigns significant probability to very few other (spurious) edges, even in the case of relatively high observation noise or if a significant portion (2/3) of data is missing. We also show that our method recovers the interaction structure better than the method of Siracusa and Fisher [50] in the case of high observation noise, as well as that our method assigns higher uncertainty to an incorrect edge in the MAP structure estimate, than to the correct ones. Lastly, we demonstrate the advantage of marginalization over switching pattern, which we employ, over the previous method that only considers a point estimate of the switching pattern.

In addition, we apply our method to real datasets. While we cannot formally test the results of switching interaction analysis on them, we see that the results are coherent with prior knowledge in the domain or general intuition. The climate indices dataset, Monthly atmospheric and ocean time series [40], consists of time-series of measurements of climate indices over several decades. Structural health monitoring (SHM) datasets are also used to perform interaction analysis. Buildings are instrumented with sensors (accelerometers) that measure vibrations at different locations. Two laboratory structures and one real building were used for experiments.

Finally, we develop extensions of the SSIM model for classification [14] and single-class classification of sequences of measurements, using an assumption that switching may only occur between sequences, and not within a sequence. These variants of the SSIM are applied to the problem of damage detection in civil buildings, which is one of the major problems in structural health monitoring. We demonstrate that our approach can detect damage or significant changes in the environment or excitation of a building with high accuracy, even in a single-class classification setup, in which only data from an intact structure is available for training (which is a typical case). The probability of a damage is in general higher for more severe damages. Also, the model can successfully differentiate different types of damages.

■ 1.3 Outline

The organization of the thesis is as follows. The necessary background material is laid in Chapter 2. The SSIM, a framework for switching interaction analysis under uncertainty, which is based on a Bayesian state-space switching structure inference, is introduced in Chapter 3, along with the Gibbs sampling inference algorithm. The

LG-SSIM, a specialization of the SSIM that uses linear-Gaussian dependence and observation models, as well as the corresponding specialization of the inference procedure, are also presented in Chapter 3. Finally, the time and memory complexity analysis of the inference algorithm is also presented here. Practical considerations regarding setting the prior and initializing the latent variables are addressed in Chapter 4. Experiments on synthetic, semi-real and real data, which demonstrate the utility of the algorithm, are also presented in Chapter 4. Chapter 5 is devoted to the application of the developed framework to the problem of damage detection in civil engineering. Finally, conclusions and directions for future work are given in Chapter 6.

Chapter 2

Background

WE take a Bayesian approach (2.1) to learning of the structure of Dynamic Bayesian networks (2.3.2), which are probabilistic graphical models (2.3) suitable for modeling time-series data. The state-space modeling paradigm is used to extend the previous work ([49, 50], summarized in 2.7) to enable inference with imperfect (noisy and missing) data. In particular, a switching state-space approach is used to model the change in structure over time, in contrast to the switching auto-regressive approach used in [49, 50]. The inference is performed using a Gibbs sampling algorithm (2.4.1), which is a Markov chain Monte Carlo (MCMC) type of algorithm (2.4). A particular choice of probability distributions with conjugate priors (2.2) used for the dependence and observation models allows for efficient Gibbs sampling steps. Overview of the Bayesian learning of a homogenous (non-switching) dependence structure is presented in Section 2.6. Efficient inference over the space of structures, which is extremely large, is enabled by the use of a modular prior and a bound on the node in-degree (2.6.2).

■ 2.1 Bayesian Approach

In contrast to the classical (or frequentist) approach, in which parameters of a statistical model are assumed fixed, but unknown, in the Bayesian approach, parameters are assumed to be drawn from some distribution (called prior distribution or simply prior) and therefore treated as random variables. Let $p(X \mid \theta)$ be a probabilistic model of a phenomenon captured by a collection of variables X, with parameters θ, and let $p(\theta; \gamma)$ be the prior distribution of model parameters θ, parametrized by γ, which are typically called hyperparameters. The prior distribution is often assumed to be known, in which case hyperparameters are treated as constants and are either chosen in advance to reflect the prior belief in the parameters θ (e.g., by a domain expert) or estimated from data (empirical Bayes, [8]). Alternatively, in a hierarchical Bayesian approach, hyperparameters are also treated as random variables and modeled via some distribution, parametrized by a next level of hyperparameters, and so on, up to some level of hierarchy.

The central computation in Bayesian inference is computing the posterior distribution of parameters θ given data samples $\mathcal{D} = \{\tilde{X}_1, \tilde{X}_2, \ldots, \tilde{X}_N\}$, namely, $p(\theta \mid \mathcal{D}; \gamma)$. If the samples are independent, the data likelihood is $p(\mathcal{D} \mid \theta) = \prod_{i=1}^{N} p(X = \tilde{X}_i \mid \theta)$. The

posterior distribution can be computed using the Bayes rule:

$$p(\theta \mid \mathcal{D}; \gamma) = \frac{p(\theta; \gamma)\, p(\mathcal{D} \mid \theta)}{p(\mathcal{D}; \gamma)} = \frac{p(\theta; \gamma)\, p(\mathcal{D} \mid \theta)}{\int_\theta p(\theta; \gamma)\, p(\mathcal{D} \mid \theta)\, d\theta}. \tag{2.1}$$

Note that the denominator $p(\mathcal{D}; \gamma)$, the marginal likelihood of data, does not depend on the parameters θ, which are "marginalized out". Therefore, the posterior distribution is proportional to the numerator:

$$p(\theta \mid \mathcal{D}; \gamma) \propto p(\theta; \gamma)\, p(\mathcal{D} \mid \theta), \tag{2.2}$$

while the denominator is simply a normalization constant.

Evaluating the numerator above for a specific value of parameters is easy, as it is the product of the prior distribution and the data likelihood terms, which are specified by the model. However, computing the full posterior distribution $p(\theta \mid \mathcal{D}; \gamma)$, or even evaluating it for a specific parameters value (which requires computing the marginal likelihood $p(\mathcal{D}; \gamma)$), is in general difficult, as the posterior distribution and the marginal likelihood may not have closed-form analytical expressions. Nonetheless, when the prior distribution, $p(\theta; \gamma)$, is chosen to be a so-called conjugate distribution to the likelihood distribution, $p(X \mid \theta)$, the posterior distribution has the same form as the prior and can be computed efficiently.

■ 2.2 Conjugate Priors

If the posterior distribution from Equation 2.1, $p(\theta \mid \mathcal{D}; \gamma)$, is in the same family as the prior distribution, $p(\theta; \gamma)$, then $p(\theta; \gamma)$ is called a **conjugate prior** for the likelihood function, $p(\mathcal{D} \mid \theta)$. In that case, we say that the probability distribution $p(\mathcal{D} \mid \theta)$ has a conjugate prior. As a consequence, if the prior distribution has a parametric form (which we will assume in this thesis) and is a conjugate prior, then the posterior distribution has the same parametric form and differs from the prior only in the value of hyperparameters, i.e., $p(\theta \mid \mathcal{D}; \gamma) = p(\theta; \gamma')$ for some γ'. Note that γ' is a function of prior hyperparameters γ and data $\mathcal{D}$. Computing γ' can be done analytically and is commonly referred to as "updating" the prior with the data or performing a "conjugate update".

Choosing a distribution that has a conjugate prior for a likelihood function and its conjugate prior for the prior is convenient as it results in an analytic form of the posterior, efficient computation of the posterior, and overall efficient inference in models that use such distributions. Otherwise, a computationally more challenging methods must be used, such as integration or sampling techniques. Also, interpreting conjugate updates is typically more intuitive than interpreting the results of numerical or sampling methods, as there is a meaning attached to the hyperparameters and how they are changed after a conjugate update.

Not all probability distributions have a conjugate prior. However, all distributions in the so-called **exponential families**, which includes a majority of well-known distributions, have a conjugate prior, and are therefore a convenient choice. We will proceed

by describing exponential families and the probability distributions that will be used in this thesis, all of which belong to the exponential family.

■ 2.2.1 Exponential Families

An **exponential family** in the case of vector-valued variable X and parameters θ (which is the case we will usually need in this thesis) is a set of probability distributions of the form:

$$p(X|\,\theta) = h(X)\,\exp\left\{\eta(\theta)^T T(X) - A(\theta)\right\}\,, \tag{2.3}$$

where $\eta(\theta)$ is referred to as the **natural parameter**, $T(X)$ as **natural statistic** or **sufficient statistic**, $h(X)$ as the **base distribution**, and $A(\theta)$ as the **cumulant function** or the **log-partition function**. Note that $\eta(\theta)$ and $T(X)$ are vectors, in general. An exponentially family is uniquely defined by the choice of $\eta(\theta)$, $T(X)$ and $h(X)$, while $A(\theta)$ is the logarithm of the normalization term implied by the previous three functions:

$$A(\theta) = \log \int_X h(X)\,\exp\left\{\eta(\theta)^T T(X)\right\} dX\,, \tag{2.4}$$

where the integral is replaced with a summation if X is a discrete variable. The normalization term, $Z(\theta) = e^{A(\theta)}$, is also called the **partition function**.

A **linear exponential family** is an exponential family whose natural parameter, $\eta(\theta)$, is equal to the underlying parameter, θ:

$$p(X|\,\theta) = h(X)\,\exp\left\{\theta^T T(X) - A(\theta)\right\}\,. \tag{2.5}$$

Note that any exponential family can be converted into a linear exponential family by changing parametrization, i.e., $p(X|\,\theta') = h(X)\,\exp\left\{\theta'^T T(X) - A(\theta')\right\}$, where $\theta' = \eta(\theta)$. However, finding the range of admissible values of θ' and the log-partition function $A(\theta')$ may pose a challenge.

A **canonical exponential family** is a linear exponential family whose natural statistics, $T(X)$, is equal to the underlying variable, X:

$$p(X|\,\theta) = h(X)\,\exp\left\{\theta^T X - A(\theta)\right\}\,. \tag{2.6}$$

Exponential families have many useful properties. For example, the log-partition function play the role of a cumulant-generating function:

$$\begin{aligned} \frac{\partial A(X)}{\partial \theta_i} &= \mathrm{E}\left[T_i(X)\right] \\ \frac{\partial^2 A(X)}{\partial \theta_i \partial \theta_j} &= \mathrm{Cov}\left[T_i(X) T_j(X)\right]\,. \end{aligned} \tag{2.7}$$

Also, the natural statistic, $T(X)$, is a **sufficient statistic**, which implies that all inferences about parameter θ can be performed using $T(X)$ – once $T(X)$ is computed, the data X can be discarded. An important property of exponential families is that the dimensionality of the sufficient statistic, $\dim(T(X))$, does not increase with the number of data samples. To see that, let $X_1, X_2, \ldots, X_N$ be independent and identically distributed (i.i.d.) random variables from a member of an exponential family defined by Equation 2.3. Then, the joint probability distribution of these variables is:

$$p(X_1, X_2, \ldots, X_N | \theta) = \left(\prod_{i=1}^{N} h(X_i) \right) \exp \left\{ \eta(\theta)^T \left(\sum_{i=1}^{N} T(X_i) \right) - NA(\theta) \right\} . \quad (2.8)$$

Note that the sufficient statistic of all samples is simply the sum of sufficient statistics of each individual variable X_i.

The property of exponential families that will be the most important for us is that **every exponential family has a conjugate prior**. If the likelihood model of joint observations is given by Equation 2.8, then

$$p(\theta; \tau, n_0) \propto \exp \left\{ \tau^T \eta(\theta) - n_0 A(\theta) \right\} \quad (2.9)$$

is a conjugate prior for that family, where τ and n_0 are hyperparameters. The posterior distribution over parameter θ is

$$p(\theta | X; \tau, n_0) \propto \exp \left\{ \left(\tau + \sum_{i=1}^{N} T(X_i) \right)^T \eta(\theta) - (n_0 + N) A(\theta) \right\} . \quad (2.10)$$

Clearly, the posterior distribution is in the same form as the prior, i.e.,

$$p(\theta | X; \tau, n_0) = p(\theta; \tau', n_0') , \quad (2.11)$$

where

$$\begin{aligned} \tau' &= \tau + \sum_{i=1}^{N} T(X_i) \\ n_0' &= n_0 + N . \end{aligned} \quad (2.12)$$

Therefore, performing a conjugate update is reduced to simply updating hyperparamters with the sufficient statistic and the sample count.

■ 2.2.2 Multinomial (Categorical) Distribution

The **multinomial distribution** is a distribution over the possible ways of selecting N items from the set of K items, with repetition. Let $\pi_1, \pi_2, \ldots, \pi_K$ be the probabilities of choosing items $1, 2, \ldots, K$, respectively. Note that $\sum_{i=1}^{K} \pi_i = 1$ must hold and N choices are made independently. Let $X_1, X_2, \ldots, X_K$ be random variables, such that

X_i correspond to the number of times item i is selected. Then, the joint probability over $X_1, X_2, \ldots, X_K$ is

$$\mathcal{M}ult(X_1, X_2, \ldots, X_k; \pi_1, \pi_2, \ldots, \pi_K) = \frac{N!}{X_1! X_2! \cdots X_K!} {\pi_1}^{X_1} {\pi_2}^{X_2} \cdots {\pi_K}^{X_K}, \quad (2.13)$$

where $X_1, X_2, \ldots, X_K$ are non-negative integers such that $\sum_{i=1}^{K} X_i = N$. This probability can also be written using the *gamma function* as

$$\mathcal{M}ult(X_1, X_2, \ldots, X_k; \pi_1, \pi_2, \ldots, \pi_K) = \frac{\Gamma\left(\sum_{i=1}^{K} X_i + 1\right)}{\prod_{i=1}^{K} \Gamma(X_i + 1)} \prod_{i=1}^{K} {\pi_i}^{X_i}, \quad (2.14)$$

which is a convenient form for a comparison to its conjugate prior – the Dirichlet distribution.

The mean and variance of a random variable X_i and covariance between X_i and X_j are given as:

$$\begin{aligned} \mathrm{E}[X_i] &= N\pi_i \\ \mathrm{Var}[X_i] &= N\pi_i(1 - \pi_i) \\ \mathrm{Cov}[X_i, X_j] &= -N\pi_i\pi_j, \quad i \neq j. \end{aligned} \quad (2.15)$$

The **categorical distribution** can be thought of as the multinomial distribution with $N = 1$, i.e.,

$$\mathcal{C}at(X_1, X_2, \ldots, X_k; \pi_1, \pi_2, \ldots, \pi_K) = \prod_{i=1}^{K} {\pi_i}^{X_i}, \quad (2.16)$$

where exactly one of the variables $X_1, X_2, \ldots, X_K$ is equal to 1, and the others are equal to 0. The categorical distribution is sometimes referred to as the **discrete distribution**, since it is a distribution over a selection of one element from a discrete set of elements, where π_i is the probability of selecting element i. It is also commonly expressed using a single random variable X that takes a value from $\{1, 2, \ldots, K\}$:

$$\mathcal{C}at(X; \pi_1, \pi_2, \ldots, \pi_K) = \prod_{i=1}^{K} {\pi_i}^{[X=i]}, \quad (2.17)$$

where $[X = i] = 1$ if $X = i$ and $[X = i] = 0$ otherwise. A connection to the representation given in Equation 2.16 is established via equality $X_i = [X = i]$. Therefore, from Equation 2.15, it follows that

$$\begin{aligned} \mathrm{E}[[X = i]] &= \mathrm{E}[X_i] = \pi_i \\ \mathrm{Var}[[X = i]] &= \mathrm{Var}[X_i] = \pi_i(1 - \pi_i) \\ \mathrm{Cov}[[X = i], [X = j]] &= \mathrm{Cov}[X_i, X_j] = -\pi_i\pi_j, \quad i \neq j. \end{aligned} \quad (2.18)$$

In machine learning, it is common to talk about a multinomial distribution when a categorical distribution is actually meant. Note also that the **binomial distribution** and the **Bernoulli distribution** are special cases of the multinomial and categorical distributions, respectively, in which the number of items, K, is equal to 2.

2.2.3 Dirichlet Prior

The **Dirichlet distribution** is a distribution over an *open* $K-1$*-dimensional simplex* in a K-dimensional space, which is defined as a set $\{(x_1, \ldots, x_K) \in \mathcal{R}^K \mid x_1 > 0, \ldots, x_K > 0, \sum_{i=1}^K x_i = 1\}$. The probability density function of a Dirichlet distribution is given by

$$\mathcal{D}ir(X_1, \ldots, X_K; \alpha_1, \ldots, \alpha_K) = \frac{1}{\mathrm{B}(\alpha_1, \ldots, \alpha_K)} {X_1}^{\alpha_1 - 1} \cdots {X_K}^{\alpha_K - 1}, \tag{2.19}$$

where $\alpha_1, \alpha_2, \ldots, \alpha_K > 0$ and $\mathrm{B}(\alpha_1, \ldots, \alpha_K)$ is the *Beta function*, which can be expressed in terms of the gamma function as:

$$\mathrm{B}(\alpha_1, \ldots, \alpha_K) = \frac{\prod_{i=1}^K \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^K \alpha_i)}. \tag{2.20}$$

The mean and variance of a random variable X_i and covariance between X_i and X_j are given as:

$$\begin{aligned} \mathrm{E}[X_i] &= \frac{\alpha_i}{\alpha_0} \\ \mathrm{Var}[X_i] &= \frac{\alpha_i(\alpha_0 - \alpha_i)}{\alpha_0^2(\alpha_0 + 1)} \\ \mathrm{Cov}[X_i, X_j] &= -\frac{\alpha_i \alpha_j}{\alpha_0^2(\alpha_0 + 1)}, \quad i \neq j, \end{aligned} \tag{2.21}$$

where $\alpha_0 = \sum_{i=1}^K \alpha_i$. Note that the mean of X does not depend on the absolute values of parameters α_i, but rather on their proportion. If all parameters α_i are scaled by a same factor, the mean does not change. However, if that factor is greater than 1 (i.e., if parameters α_i increase proportionally) and assuming that initially $\alpha_i \geq 1$, $\forall i$, the variance of each X_i decreases, meaning that the distribution on $X_1, \ldots, X_K$ becomes narrower around the mean.

Note that the support of the Dirichlet distribution is also the domain of possible distributions over a discrete set of K elements. Furthermore, the Dirichlet distribution is a conjugate prior to the multinomial (categorical) distribution. If the likelihood model is given by Equation 2.13 and the prior on parameters $\pi_1, \ldots, \pi_K$ as

$$\mathcal{D}ir(\pi_1, \ldots, \pi_K; \alpha_1, \ldots, \alpha_K) = \frac{\Gamma(\sum_{i=1}^K \alpha_i)}{\prod_{i=1}^K \Gamma(\alpha_i)} {\pi_1}^{\alpha_1 - 1} \cdots {\pi_K}^{\alpha_K - 1}, \tag{2.22}$$

and the observed valued are $X_1 = c_1, \ldots, X_K = c_K$, then the posterior probability of parameters is

$$\mathcal{D}ir(\pi_1, \ldots, \pi_K; \alpha'_1, \ldots, \alpha'_K) = \frac{\Gamma(\sum_{i=1}^K \alpha'_i)}{\prod_{i=1}^K \Gamma(\alpha'_i)} {\pi_1}^{\alpha'_1 - 1} \cdots {\pi_K}^{\alpha'_K - 1}, \tag{2.23}$$

where $\alpha'_1 = \alpha_1 + c_1, \ldots, \alpha'_K = \alpha_K + c_K$. Therefore, a Dirichlet conjugate update is performed by simply updating each parameter α_i with the number of samples from category i, c_i. Parameters α_i are also called pseudo counts as they are added to the observed counts. Having a prior parameter α_i is equivalent to having a prior parameter $\alpha_i - d_i$ and adding d_i pseudo observations from category i. Note from Equation 2.21 that the proportion of parameters α_i determines the mean of probabilities π_i, while their magnitude determines the variance of probabilities π_i and thus reflects the strength of belief in the mean values. In general, the larger α_i values are (the more pseudo-observations there are), the narrower the distribution on π_i parameters is, meaning that the belief is stronger. Conversely, small values of α_i parameters result in a prior with large variance, which is referred to as a weak (or broad) prior.

■ 2.2.4 Normal Distribution

The **(multivariate) normal distribution**, also called the **(multivariate) Gaussian distribution**, is a distribution over d-dimensional real vectors, $X = [X_1 X_2 \ldots X_d]^T$, with a density function

$$\mathcal{N}(X; \mu, \Sigma) = \frac{\exp(-\frac{1}{2}(X-\mu)^T \Sigma^{-1}(X-\mu))}{(2\pi)^{d/2}|\Sigma|^{1/2}}, \tag{2.24}$$

where μ is a d-dimensional vector and Σ is a positive definite matrix of size $d \times d$, which are also the mean and covariance matrix of X, respectively. I.e.,

$$\begin{aligned} \mathrm{E}[X] &= \mu \\ \mathrm{Cov}[X] &= \Sigma, \end{aligned} \tag{2.25}$$

which is a shorthand for the set of equalities

$$\begin{aligned} \mathrm{E}[X_i] &= \mu_i \\ \mathrm{Var}[X_i] &= \Sigma_{ii} \\ \mathrm{Cov}[X_i, X_j] &= \Sigma_{ij}. \end{aligned} \tag{2.26}$$

A conjugate prior to the normal distribution with a known covariance matrix is also a normal distribution. If the likelihood models is given as

$$p(X \mid \mu; \Sigma) = \mathcal{N}(X; \mu, \Sigma) \tag{2.27}$$

and the prior on μ as

$$p(\mu; \mu_0, \Sigma_0) = \mathcal{N}(\mu; \mu_0, \Sigma_0), \tag{2.28}$$

and there are n independent samples of variable X, $x_1, \ldots, x_n$, then the posterior distribution of the mean μ is

$$p(\mu \mid x_1, \ldots, x_n; \mu_0, \Sigma_0) = \mathcal{N}(\mu; \mu'_0, \Sigma'_0), \tag{2.29}$$

where

$$\begin{aligned}\mu'_0 &= \left(\Sigma_0{}^{-1} + n\,\Sigma^{-1}\right)^{-1}\left(\Sigma_0{}^{-1}\mu_0 + n\,\Sigma^{-1}\bar{x}\right)\\ \Sigma'_0 &= \left(\Sigma_0{}^{-1} + n\,\Sigma^{-1}\right)^{-1},\end{aligned} \tag{2.30}$$

and $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ is the sample mean.

■ 2.2.5 Inverse-Wishart Prior

The **inverse-Wishart distribution** is a distribution over positive-definite matrices of a fixed dimension, $d \times d$, with a density function

$$\mathcal{IW}(X;\Psi,\kappa) = \frac{|\Psi|^{\kappa/2}}{2^{\kappa d/2}\Gamma_d(\kappa/2)}\,|X|^{-(\kappa+d+1)/2}\exp(-\frac{1}{2}\operatorname{tr}(\Psi X^{-1}))\,, \tag{2.31}$$

where $\Gamma_d()$ is the *multivariate gamma function* [30], $\kappa > d-1$ is a scalar parameter called the degrees of freedom, and Ψ is a $d \times d$ positive definite matrix parameter called the inverse scale matrix.

The mean and the mode of an inverse-Wishart distributed random matrix X are not equal:

$$\begin{aligned}\mathrm{E}[X] &= \frac{\Psi}{\kappa-d-1}\,, \quad \kappa > d+1\\ \mathrm{Mode}[X] &= \frac{\Psi}{\kappa+d+1}\,.\end{aligned} \tag{2.32}$$

For larger values of κ the variance of X is smaller, and therefore the distribution is narrower around the mode.

The inverse-Wishart distribution is a conjugate prior to the normal distribution with a known mean. If the likelihood models is given as

$$p(X \mid \Sigma;\, \mu) = \mathcal{N}(X;\mu,\Sigma) \tag{2.33}$$

and the prior on Σ as

$$p(\Sigma;\, \Psi,\kappa) = \mathcal{IW}(\Sigma\,;\Psi,\kappa)\,, \tag{2.34}$$

and there are n independent samples of variable X, $x_1,\ldots,x_n$, then the posterior distribution of the covariance matrix Σ is

$$p(\Sigma \mid x_1,\ldots,x_n\,;\Psi,\kappa) = \mathcal{IW}(\Sigma;\Psi',\kappa')\,, \tag{2.35}$$

where

$$\begin{aligned}\Psi' &= \Psi + \sum_{i=1}^{n}(x_i-\mu)(x_i-\mu)^T\\ \kappa' &= \kappa + n\,.\end{aligned} \tag{2.36}$$

Note that setting a small value of κ defines a weak (broad) prior on Σ, and that κ can also be thought of as a pseudo-count.

■ 2.2.6 Matrix-Normal Inverse-Wishart Prior

Here, we consider a linear Gaussian model of a multivariate signal X_t,

$$X_t = AX_{t-1} + w_t\,, \qquad w_t \sim \mathcal{N}(0, Q)\,, \tag{2.37}$$

with parameters A (transition matrix) and Q (noise covariance matrix).

We assume that $\Theta = (A, Q)$ follows a matrix-normal inverse-Wishart distribution, which is a conjugate prior to the dependence model $\mathcal{N}(X_t; AX_{t-1}, Q)$:

$$\begin{aligned} p(A, Q; M, \Omega, \Psi, \kappa) &= \mathcal{MN}\text{-}\mathcal{IW}(A, Q; M, \Omega, \Psi, \kappa) \\ &= \mathcal{MN}(A; M, Q, \Omega)\,\mathcal{IW}(Q; \Psi, \kappa)\,. \end{aligned} \tag{2.38}$$

It is a product of (1) the matrix-normal distribution

$$\mathcal{MN}(A; M, Q, \Omega) = \frac{\exp\left(-\frac{1}{2}\operatorname{tr}\left[\Omega^{-1}(A-M)^T Q^{-1}(A-M)\right]\right)}{(2\pi)^{dl/2}|\Omega|^{d/2}|Q|^{l/2}}\,, \tag{2.39}$$

where d and l are the dimensions of matrix A ($A_{d\times l}$), while $M_{d\times l}$, $Q_{d\times d}$ and $\Omega_{l\times l}$ are the mean, the column covariance and the row covariance parameters; and (2) the inverse-Wishart distribution

$$\mathcal{IW}(Q; \Psi, \kappa) = \frac{|\Psi|^{\kappa/2}}{2^{\kappa d/2}\Gamma_d(\kappa/2)}\,|Q|^{-(\kappa+d+1)/2}\exp(-\frac{1}{2}\operatorname{tr}(\Psi Q^{-1}))\,, \tag{2.40}$$

where d is the dimension of matrix Q ($Q_{d\times d}$) and $\Gamma_d()$ is a multivariate gamma function while κ and $\Psi_{d\times d}$ are the degree of freedom and the inverse scale matrix parameters. Note how the two distributions are coupled. The matrix normal distribution of the dependence matrix A depends on the covariance matrix Q, which is sampled from the inverse Wishart distribution.

Due to conjugacy, the posterior distribution of parameters A and Q given data sequence $X_0, X_1, \ldots, X_T$ is also a matrix-normal inverse-Wishart distribution:

$$\begin{aligned} p(A, Q|X_{0:T}; M, \Omega, \Psi, \kappa) &= \mathcal{MN}\text{-}\mathcal{IW}(A, Q; M', \Omega', \Psi', \kappa') \\ &= \mathcal{MN}(A; M', Q, \Omega')\,\mathcal{IW}(Q; \Psi', \kappa')\,, \end{aligned} \tag{2.41}$$

where

$$\begin{aligned} \Omega' &= \left(\Omega^{-1} + \sum_{t=0}^{T-1} X_t X_t^T\right)^{-1} \\ M' &= \left(M\Omega^{-1} + \sum_{t=1}^{T} X_t X_{t-1}^T\right)\Omega' \\ \kappa' &= \kappa + T \\ \Psi' &= \Psi + \sum_{t=1}^{T} X_t X_t^T + M\Omega^{-1}M^T - M'\Omega'^{-1}M'^T\,. \end{aligned} \tag{2.42}$$

2.3 Graphical Models

In general, full representation of a probability distribution over N variables is intractable. If these are discrete variables that take value from a set with cardinality S, the full representation of an arbitrary joint distribution among them requires $S^N - 1$ parameters, as there is freedom in setting the probability for each combination of variable values except for one (since they have to sum to 1). Similarly, for continuous variables, the most general representation would require infinitely many parameters. Therefore, only distributions that can be represented compactly are used in practice, as is the case with all known families of distributions. In probabilistic modeling, a probability mass or distribution function is typically assumed to take some parametric form with a finite number of parameters. For example, a (multivariate) Gaussian distribution over N univariate continuous variables, $\mathcal{N}(\cdot\,; \mu, \Sigma)$, is represented with mean μ, which is a vector of length N, and covariance matrix Σ of size $N \times N$. In another example, let us assume that D is a discrete random variable that takes a value from $\{1, 2, \ldots, K\}$ and is distributed according to a multinomial distribution with parameters $\pi = (\pi_1, \pi_2, \ldots, \pi_K)$, $(\pi_i \geq 0, \sum_{i=1}^{K} \pi_i = 1)$, while π itself is a multivariate random variable that is distributed according to a Dirichlet distribution with parameters $\alpha = (\alpha_1, \alpha_2, \ldots . \alpha_K)$. Then, the joint distribution of π and D can be written as $p(\pi, D; \alpha) = p(\pi; \alpha)\, p(D \mid \pi) = Dirichlet(\pi; \alpha) Mult(D; \pi)$.

Graphical models are a language that uses graphs to compactly represent families of joint probability distributions among multiple variables that respect certain constraints dictated by a graph. There are two common types: undirected graphical models (also called Markov random fields) and directed graphical models (Bayesian networks), which use undirected and acyclic directed graphs, respectively, to form such constraints. In both cases, nodes of a graph correspond to the variables which joint distribution is modeled. In an undirected graphical model, a joint probability distribution is assumed to be proportional to a product of nonnegative functions (called potentials) over graph cliques (fully connected subgraphs). In a Bayesian network, a distribution is assumed to be a product of conditional distributions of each variable given its parents in the graph. Examples of both types of graphical models are shown in Figure 2.1. In both types of models, a distribution is represented as or proportional to a product of factors – potentials in undirected and conditional distributions in directed model. While each factor may still require some compact representation, such as a parametric function or a table of values (in discrete case), the complexity of this representation depends on the size of a factor (i.e., the number of variables involved in a factor) instead of on the total number of variables. Therefore, the overall complexity of a graphical model representation (and consequently inference algorithms) is typically dominated by large factors.

(a) (b)

Figure 2.1: (a) Undirected graphical model example: $P(A,B,C,D,E) \propto f_1(A,B)\, f_2(A,C)\, f_3(B,D)\, f_4(C,D)\, f_5(B,D,E)$. (b) Directed graphical model example: $P(A,B,C,D,E) = P(A)\, P(B|A)\, P(C)\, P(D|A,B,C)\, P(E|B,D)$.

■ 2.3.1 Directed Graphical Models (Bayesian Networks)

A Bayesian network (BN) consists of a directed acyclic graph $G = (V,E)$, whose nodes $X_1, X_2, \ldots, X_N$ represent random variables, and a set of conditional distributions $p(X_i|\, pa(X_i))$, $i = 1, \ldots, N$, where $pa(X_i)$ is a set of variables that correspond to the parent nodes (parents) of node X_i. Since G is acyclic, its nodes can be arranged in a so-called topological order, such that all parents of a node are its predecessors in the topological order (i.e., all edges go from left to right with respect to the topological order). Let's assume, without loss of generality, that $X_1, X_2, \ldots, X_N$ is a topological order of nodes in graph G. Then, $pa(X_i) \in \{X_1, X_2, \ldots, X_{i-1}\}$. Note that any joint distribution among N variables can be written as

$$p(X_1, X_2, \ldots, X_N) = p(X_1)\, p(X_2|X_1) \ldots p(X_N|X_1, \ldots, X_{N-1}) = \prod_{i=1}^{N} p(X_i|X_1, \ldots, X_{i-1})\,.$$

A Bayesian network with associated graph G represents a family of distributions of the form

$$p(X_1, X_2, \ldots, X_N) = \prod_{i=1}^{N} p(X_i|\, pa(X_i))\,,$$

i.e., in which each variable X_i, when conditioned on its parents $pa(X_i)$, is independent of all other predecessors in a topological sort $(X_i \perp\!\!\!\perp \{X_1, \ldots, X_{i-1}\} \setminus pa(X_i) \,|\, pa(X_i))$.[1] Conditional distributions $p(X_i|\, pa(X_i))$ are typically assumed to have some parametric form $p(X_i|\, pa(X_i), \theta_i)$, in which case learning a Bayesian network means learning parameters θ_i. If, in addition, graph G is unknown, the inference of this graph is commonly referred to as learning the structure of a Bayesian network.

Figure 2.2 shows two additional examples of Bayesian networks. In Figure 2.2a, $D_1, D_2, \ldots, D_N$ are discrete random variables with values from $\{1, 2, \ldots, K\}$ that are drawn independently from a multinomial distribution with parameters $\pi = (\pi_1, \pi_2, \ldots, \pi_K)$, $(\pi_i \geq 0, \sum_{i=1}^{K} \pi_i = 1)$, while π itself is a random vector drawn from a Dirichlet distribution with parameters $\alpha = (\alpha_1, \alpha_2, \ldots . \alpha_K)$. Then, the overall joint distribution can

[1] There can be multiple topological sorts for the same graph. This holds for any of them.

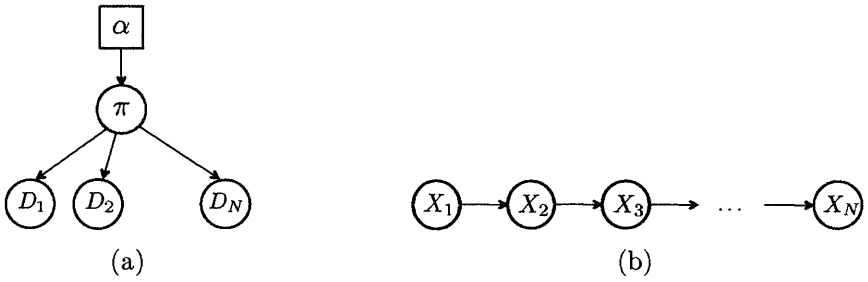


Figure 2.2: Two examples of Bayesian networks.

be written as

$$p(\pi, D_1, D_2, \ldots, D_N; \alpha) = p(\pi; \alpha) \prod_{i=1}^{N} p(D_i \mid \pi) = Dirichlet(\pi; \alpha) \prod_{i=1}^{N} Mult(D_i; \pi) .$$

Note that if constant parameters are shown in a graphical model diagram (α in this case), they are written inside a square (as here) or simply without an associated graphical symbol. In Figure 2.2b, $X_1, X_2, \ldots, X_N$ are jointly Gaussian univariate random variables with an additional constraint that, for each i, X_i is independent of $X_1, \ldots, X_{i-2}$ when conditioned on X_{i-1} (first order Markov assumption):

$$P(X_1, X_2, \ldots, X_N) = P(X_1) \prod_{i=2}^{N} P(X_i|X_{i-1}) = \mathcal{N}(X_1; \mu_1, \sigma_1^2) \prod_{i=2}^{N} \mathcal{N}(X_i; a_i X_{i-1}, \sigma_i^2) .$$

Note that this model requires only $2N$ parameters, compared to $N + N^2$ required for a general multivariate Gaussian model. If for example, parameters a_i and σ_i^2 are assumed the same for all i, the number of parameters is further reduced to 3.

■ 2.3.2 Temporal Directed Graphical Models (Dynamic Bayesian Networks)

Dynamic Bayesian networks (DBNs) are Bayesian networks that model sequential data, such as time-series. In a DBN, random variables are indexed with discrete numbers $0, 1, 2, \ldots, T$ (we choose to start with 0 for convenience, but starting index can be arbitrary). We will refer to such index as time, although it may not be time-related in general (for example, it can be an index into a genome sequence or a word in a sentence). Each signal in a model is therefore represented with a sequence of random variables that correspond to its value at different indices, or discrete time points. Edges are allowed only from a variable with a lower index to a variable with a higher index (i.e., they must "point" forward in time). Let X_t^i denote a random variables that takes the value of signal i at time t. Then, if there is an edge from $X_{t_1}^i$ to $X_{t_2}^i$, $t_2 > t_1$ must hold. Furthermore, edges are often restricted to connect variables at neighboring time points, i.e., they are

of the form $X_t^i \rightarrow X_{t+1}^j$. This assumption results in a first-order Markov model over time – signal values at time t are independent of the past given their values at time $t-1$. We will assume such models throughout this paper. Let $pa(i,t)$ be the set of parents of signal i at time t. Then, the associated conditional probability distributions are of the form $p(X_t^i \,|\, X_{t-1}^{pa(i,t)})$, where $X_{t-1}^{pa(i,t)}$ denotes a collection of variables $\{X_{t-1}^v;\, v \in pa(i,t)\}$. In homogenous DBNs (which are often assumed by the term DBN) edges between signals (i.e., parent sets) and conditional distributions are assumed time-invariant. On the other hand, in time-varying DBNs both edges and conditional distributions may vary over time. Figure 1.1 shows an example of a time-varying DBN which is piecewise homogenous (switching).

■ 2.4 Markov Chain Monte Carlo Sampling

Markov chain Monte Carlo (MCMC) sampling is a class of algorithms for generating samples from a distribution $p^*(x)$ via a random walk on a Markov chain that has distribution $p^*(x)$ as its stationary distribution. A **Markov chain** is a stochastic process defined as a sequence of random variables $X_1, X_2, X_3, \ldots$ that satisfy Markov property:

$$p(X_n \,|\, X_{n-1}, X_{n-2}, \ldots, X_1) = p(X_n \,|\, X_{n-1}), \quad \forall n > 1\,. \tag{2.43}$$

In other words, the value of the random variable X_n depends only on the value of the previous random variable, X_{n-1}. $p(X_n \,|\, X_{n-1})$ is a distribution of X_n given X_{n-1}, which is referred to as the transition distribution from time $n-1$ to time n. A **homogenous Markov chain** is a Markov chain for which the transition distribution is the same over time, i.e., $p(X_n = y \,|\, X_{n-1} = x) = p(X_{n-1} = y \,|\, X_{n-2} = x)$, $\forall n > 1$. Let $q(y \,|\, x)$ be this distribution, and let $\mathcal{X}$ be the domain of variables X_n, which is also called the state space of a Markov chain. Then, a homogenous Markov chain is described by the state space $\mathcal{X}$ and a transition distribution $q(y \,|\, x)$, $\forall x, y \in \mathcal{X}$. In the rest of this Section we will assume that Markov chains are homogenous.

A **stationary distribution** of a homogenous Markov chain is a distribution $p^*(\cdot)$ over the state space that is invariant under the transition distribution:

$$p^*(y) = \int_{x \in \mathcal{X}} p^*(x)\, q(y \,|\, x)\, dx\,, \quad \forall y \in \mathcal{X}\,. \tag{2.44}$$

State y is *reachable* from state x if there exists $n > 0$ such that $p(X_n = y \,|\, X_1 = x) > 0$ (i.e., if there exists a sequence of transitions that reach state y from state x. State x is *aperiodic* if there exists n_0 such that $p(X_n = x \,|\, X_1 = x) > 0$ $\forall n \geq n_0$. If all states are aperiodic and reachable from each other, a Markov chain is said to be *ergodic* and converges to a unique stationary distribution starting from any state X_1. In other words, a distribution $p(X_n)$ will become closer to the stationary distribution as n approaches infinity.

The convergence property of ergodic Markov chains is exploited in the MCMC sampling approach. Namely, if the transition distribution of a Markov chain is defined in such a way that the chain is ergodic and the stationary distribution is equal to a target distribution $p^*(\cdot)$, then the Markov chain converges to the target distribution. Samples from the target distribution are generated by simulating the Markov chain.

There are several practical considerations regarding MCMC methods. The target distribution is never truly achieved in finite number of steps. However, after certain number of transitions, n_0, $p(X_{n_0})$ becomes close enough to the target distribution $p^*(\cdot)$ that it can be assumed equal to $p^*(\cdot)$ for practical purposes. The same then hold for any $n \geq n_0$. But, the question is how big n_0 should be? That depends on a particular application and is typically estimated empirically. The time n_0 after which the distribution of X_n can be assumed equal to the target distribution is called *burn-in period*. In addition, samples generated via Markov chain are correlated. The correlation between two samples is higher when they are closer to each other in the chain. Therefore, to obtain approximately independent samples from the target distribution, they should be taken at some distance apart from each other. Finally, if the target distribution is multimodal, a sampler may "get stuck" in one of the modes for a very long time. To explore the entire space more efficiently, multiple simulations of a Markov chain with different (random) initial states are often performed, and a number of samples are taken from each chain. That reduces bias towards a particular subspace of the state space.

To complete the MCMC sampling method, it remains, for a given target distribution $p^*(\cdot)$, to find a transition distribution $q(y \mid x)$ that defines an ergodic Markov chain, and for which $p^*()$ is the stationary distribution. One possible approach is to find a transition distribution that satisfies the **detailed balance**:

$$p^*(x)\, q(y \mid x) = p^*(y)\, q(x \mid y), \quad \forall x, y \in \mathcal{X}. \tag{2.45}$$

If this equation is satisfied, $p^*(\cdot)$ is guaranteed to be a stationary distribution of a Markov chain.

■ 2.4.1 Gibbs Sampling

Gibbs sampling is an MCMC sampling method that is used for sampling from a joint distribution of variables $X_1, X_2, \ldots, X_N$ when direct sampling from the joint distribution is difficult, but sampling from conditional distributions $p(X_i \mid X_{-i})$ is feasible, where X_{-i} denotes the collection of all variables except X_i. The following transition distribution is used. Let $x_1, x_2, \ldots, x_N$ be the current state. Index i is drawn randomly, and a value x'_i is sampled from the conditional distribution $p(X_i \mid X_{-i} = x_{-i})$. The new state is then $x_1, \ldots, x_{i-1}, x'_i, x_{i+1}, \ldots, x_N$. It can be shown that the transition distribution implied by this procedure satisfies the detailed balance. Furthermore, instead of drawing a value of index i randomly, i can loop through all indices in a deterministic fashion, and that procedure still converges to the stationary distribution, which is in this case the joint distribution over $X_1, X_2, \ldots, X_N$.

■ 2.5 Interaction graphs and DBN

Our goal is to reason over time-varying interactions (dependence structures) between N multivariate signals. We assume that signals evolve according to a Markov process over discrete time points $t = 0, 1, \ldots, T$. The latent state associated with signal i at time point $t > 0$ depends on the state of a subset of signals $pa(i,t)$ at time point $t-1$. We refer to $pa(i,t)$ as a parent set of signal i at time point t. While the preceding implies a first-order Markov process, the approach extends to higher-ordered Markov processes. A collection of directed edges $E_t = \{(v,i);\ i = 1,\ldots,N,\ v \in pa(i,t)\}$ forms an interaction graph at time point t, $G_t = (V, E_t)$, where $V = \{1,\ldots,N\}$ is the set of all signals. That is, there is an edge from j to i in G_t if and only if signal i at time point t depends on signal j at time point $t-1$. We say that the parent signals $pa(i,t)$ influence signal i at time t.

Let X_t^i denote a (multivariate) random variable that describes the latent state associated to signal i at time point t. Then, signal i depends on its parents at time t according to a probabilistic model $p(X_t^i | X_{t-1}^{pa(i,t)}, \theta_t^i)$ parametrized by θ_t^i, where $X_{t-1}^{pa(i,t)}$ denotes a collection of variables $\{X_{t-1}^v;\ v \in pa(i,t)\}$. Furthermore, we assume that conditioned on their parents at the previous time point, signals are independent of each other:

$$p(X_t | X_{t-1}, E_t, \theta_t) = \prod_{i=1}^{N} p(X_t^i | X_{t-1}^{pa(i,t)}, \theta_t^i)\ , \tag{2.46}$$

where $X_t = \{X_t^i\}_{i=1}^N$ (i.e., X_t is a collection of variables of all signals at time point t) and $\theta_t = \{\theta_t^i\}_{i=1}^N$. Structure E_t and parameters θ_t determine a dependence model at time t, $\mathcal{M}_t = (E_t, \theta_t)$. Finally, we express a joint probability of all variables at all time points, X, as

$$\begin{aligned} p(X) &= p(X_0|\theta_0) \prod_{t=1}^{T} p(X_t | X_{t-1}, E_t, \theta_t) \\ &= \prod_{i=1}^{N} p(X_0^i|\theta_0^i) \prod_{t=1}^{T} \prod_{i=1}^{N} p(X_t^i | X_{t-1}^{pa(i,t)}, \theta_t^i). \end{aligned} \tag{2.47}$$

The stochastic process of Eq. 2.47 can be represented using a dynamic Bayesian network (DBN), such that there is a one-to-one correspondence between the network and the collection of interaction graphs over time, as shown in Figure 1.1.

■ 2.6 Bayesian Learning of a Time-Homogenous Dependence Structure

Even when the dependence model does not change over time and observations are assumed perfect, learning a dependence structure is an NP-hard problem in general [10]. On the other hand, if we cannot solve this problem, we have little hope of solving a more complex problem of reasoning over time-changing interaction from imperfect data stated

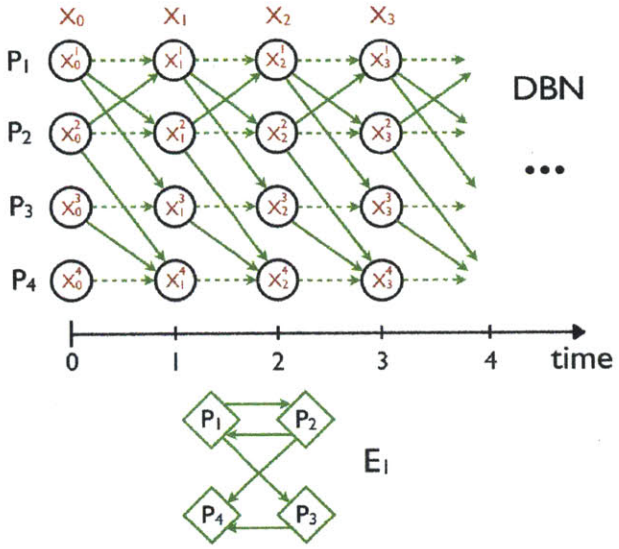


Figure 2.3: Dynamic Bayesian Network (DBN) representation of a homogenous interaction among four signals with interaction graph E. Self-edges are assumed.

in this thesis. Therefore, it is of critical importance that we have a tractable solution to the simplified problem. Furthermore, the inference over homogenous dependence structure from "perfect" data will serve as one step in the inference procedure for the full model, given in Section 3.4.

Let us first describe the homogenous interaction model more formally following the notation introduced in Section 2.5. We assume here that the dependence model is homogenous in time, i.e., $E_t \equiv E$, $pa(i,t) \equiv pa(i)$, and $\theta_t \equiv \theta$. Equation 2.46 can now be rewritten as

$$p(X_t|X_{t-1}, E, \theta) = \prod_{i=1}^{N} p(X_t^i|X_{t-1}^{pa(i)}, \theta^i) \,, \tag{2.48}$$

and Equation 2.47 as

$$\begin{aligned} p(X|E,\theta) &= p(X_0|\theta_0) \prod_{t=1}^{T} p(X_t|X_{t-1}, E, \theta) \\ &= \prod_{i=1}^{N} p(X_0^i|\theta_0^i) \prod_{t=1}^{T} \prod_{i=1}^{N} p(X_t^i|X_{t-1}^{pa(i)}, \theta^i) \,. \end{aligned} \tag{2.49}$$

This stochastic process is illustrated in Figure 2.3. In the rest of this section, we will assume that the parameters of the initial model, θ_0, are known, and focus solely on inference over the dependence model.

The goal of structure learning is to infer the dependence structure E from observed time-series X. Parameters of the dependence model, θ, may be inferred as well, or

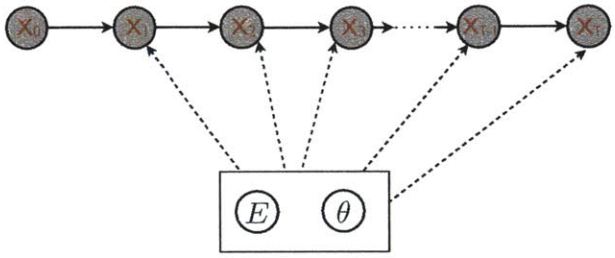


Figure 2.4: Frequentist homogenous temporal interaction model.

treated as nuisance variables. There are different approaches to structure learning, which are summarized in Section 2.6.5. We make two important distinctions here. The first one is between frequentist and Bayesian approaches. The second distinction is between point estimation of a structure and a evaluating the full posterior distribution over structures.

■ 2.6.1 Frequentist vs. Bayesian approach

Frequentist approach

In a **frequentist approach**, unknown variables are treated as deterministic (just unknown). The graphical model of a homogenous interaction in a frequentist approach is shown in Figure 2.4. In this case, the structure E and parameters θ of the dependence model (Equations 2.48 and 2.49) are unknown. The box around these two variables signifies that they are treated as a single "unit" (which we also call a dependence model), and each variable X_t depends on both of them (so, there is no need to clutter the figure by drawing a separate edge from E to X_t and from θ to X_t).

Bayesian approach

On the other hand, in a **Bayesian approach**, unknown variables are treated as random variables whose values are assumed to be generated from some prior distribution (prior to data generation). Let $p(E;\beta)$ be the prior probability of structure E, parameterized by β. In the most general form, β is a collection of parameters $\{\beta_E\}$ (one parameter for each structure), such that β_E is proportional to the prior probability of E:

$$p(E;\beta) = \frac{1}{B}\beta_E \propto \beta_E\,, \tag{2.50}$$

where $B = \sum_E \beta_E$ is a normalization constant.

Let $p(\theta\,|E;\gamma)$ be the prior probability of θ, parameterized by γ. For now, we do not assume any particular form of the dependence models, $p(X_t^i|X_{t-1}^{pa(i)},\theta^i)$. Note however that the prior on parameters, θ, may depend on the structure. Since different structures may differ in the number of parents (for some signals), they may also require

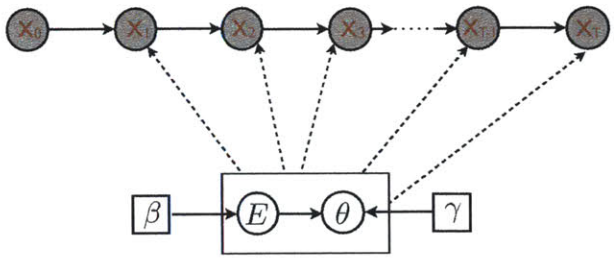


Figure 2.5: Bayesian homogenous temporal interaction model.

parameters of different dimensionality. Thus, γ is indeed a collection $\{\gamma_E\}$ of sets of hyperparameters, such that $p(\theta \,|E;\gamma) = p(\theta;\gamma_E)$.

The Bayesian model described above is shown in Figure 2.5. The posterior distribution over dependence model structure and parameters can be written as

$$p(E,\theta \,|\, X;\beta,\gamma) = \frac{p(X \,|E,\theta)\, p(E,\theta;\beta,\gamma)}{p(X;\beta,\gamma)}\,. \tag{2.51}$$

Here, $p(E,\theta;\beta,\gamma) = p(E;\beta)\, p(\theta \,|E;\gamma)$ is the joint prior on E and θ. The denominator $p(X;\beta,\gamma)$, which serves as a normalization constant, is the marginal probability of data:

$$p(X;\beta,\gamma) = \sum_E \int_\theta p(X \,|E,\theta)\, p(E,\theta;\beta,\gamma)\, d\theta\,. \tag{2.52}$$

Note that for discrete θ, the integral above should be replaced by a summation. Also, if only some components of θ are discrete, there would be a combination of a sum and an integral instead.

Similarly as with the prior, the posterior in Equation 2.51 can be decomposed as a product of the posterior over structure and the posterior over parameters given structure:

$$p(E,\theta \,|\, X;\beta,\gamma) = p(E \,|\, X;\beta,\gamma)\, p(\theta \,|\, E,X;\gamma)\,. \tag{2.53}$$

The posterior over structure can be obtained as

$$p(E \,|\, X;\beta,\gamma) = \frac{p(E;\beta)\, p(X \,|E;\gamma)}{p(X;\beta,\gamma)}\,. \tag{2.54}$$

Here, $p(X \,|E;\gamma)$ is the marginal probability of data given structure E, where the marginalization is over parameters θ:

$$p(X \,|E;\gamma) = \int_\theta p(X,\theta \,|E;\gamma)\, d\theta = \int_\theta p(X \,|E,\theta)\, p(\theta \,|E;\gamma)\, d\theta\,. \tag{2.55}$$

Note that $p(X \,|E;\gamma)$ depends on γ – the hyperparameters associated with the prior on θ. More precisely, it depends on γ_E, which are the hyperparameters associated with the

prior on θ given E. Thus, we can write $p(X\,|E;\gamma) = p(X\,|E;\gamma_E)$. If one marginalizes over the parameters θ, then Eq. 2.52 can be written as

$$p(X;\beta,\gamma) = \sum_E p(X,E;\beta,\gamma) = \sum_E p(E;\beta)\, p(X\,|E;\gamma_E)\,. \tag{2.56}$$

Finally, the posterior over parameters θ given structure can be written as

$$p(\theta\,|\,E,X;\gamma) = \frac{p(\theta\,|E;\gamma)\, p(X\,|E,\theta)}{p(X\,|E;\gamma)}\,. \tag{2.57}$$

Note that $p(X\,|E;\gamma)$, the marginal probability of data given structure, serves as a normalization constant when evaluating the posterior over parameters given structure (Eq. 2.57), while it has the role of a likelihood function when evaluating the posterior over structure (Eq. 2.54).

■ 2.6.2 Point estimation vs. full posterior distribution evaluation

Point estimation

A **point estimate** of a structure is commonly obtained as a structure that maximizes some objective function exactly or approximately (e.g., using a heuristic search) [11].

For example, a **maximum likelihood (ML) estimate** of a homogenous structure is obtained as

$$\hat{E}_{ML} = \arg\max_E \max_\theta p(X|E,\theta)\,. \tag{2.58}$$

The problem of structure learning can also be thought of as a model selection problem. For each structure E, $p(X|E,\theta)$ represents a statistical model of time series X – the one indexed by E, parametrized by θ. The maximum likelihood estimate of parameters of this model is obtained as

$$\hat{\theta}_{ML\,|\,E} = \arg\max_\theta p(X|E,\theta)\,. \tag{2.59}$$

Thus, the ML estimate of a structure is the structure that yields a model for which the highest likelihood is achieved:

$$\hat{E}_{ML} = \arg\max_E p(X|E,\hat{\theta}_{ML\,|\,E})\,. \tag{2.60}$$

In general, this criterion may lead to severe overfitting. For example, let structures E_1 and E_2 satisfy $E_1 \subset E_2$, and models $p(X|E_1,\theta)$ and $p(X|E_2,\theta)$ be such that the model for E_2 "contains" the model for E_1. In other words, $\forall\theta_1 \in \Omega_1, \exists\theta_2 \in \Omega_2$ such that $p(X|E_2,\theta_2) \equiv p(X|E_1,\theta_1)$,[2] where Ω_1 and Ω_2 are the parameter spaces of the first and second model, respectively. Then, $p(X|E_1,\hat{\theta}_{ML\,|\,E_1}) \leq p(X|E_2,\hat{\theta}_{ML\,|\,E_2})$ necessarily

[2] i.e., $\forall X, p(X|E_2,\theta_2) = p(X|E_1,\theta_1)$

holds. This is often the case in practice. For example, if the two models are chosen from the same parametric family, the model for E_2 reduces to the model for E_1 when the edges in $E_2 \setminus E_1$ are ignored (e.g., when the corresponding parameters are set to 0, or in some other way, depending on the actual family). Thus, the model for E_2 is at least as good of a fit as the model for E_1, and likely better, which results in selecting a maximal structure (fully-connected graph) as an ML estimate, and, consequently, overfitting.

A penalty on model complexity is typically imposed in order to prevent overfitting. Two commonly used objectives that incorporate model complexity are Aikike information criterion (AIC) [1] and Bayesian information criterion (BIC) [48]. AIC value of a model is defined as

$$AIC = 2m - 2\ln(L)\,, \tag{2.61}$$

where L is the maximized value of data likelihood under that model, and m is the number of independent parameters of the model. AIC criterion, which has an information-theoretic justification, states that the model with the smallest AIC value should be selected. While the negative log-likelihood generally decreases with the model complexity, the number of parameters on the other hand increases, thus providing a penalty on model complexity. In the homogenous-structure learning problem, the maximized likelihood of a model for a given structure E is

$$L(E) = p(X|E, \hat{\theta}_{ML\,|\,E})\,, \tag{2.62}$$

and Aikike information criterion is then

$$AIC(E) = 2m(E) - 2\ln p(X|E, \hat{\theta}_{ML\,|\,E})\,, \tag{2.63}$$

where $m(E)$ is the number of independent parameters of the model $p(X|E,\theta)$ (i.e., the true dimensionality of θ_E).

The BIC value of a model is defined as

$$BIC = m\ln(T) - 2\ln(L)\,, \tag{2.64}$$

where T is the number of data points, and m and L are as above. It is derived as an approximation to the marginal data likelihood assuming a "flat" prior on parameters (i.e., assuming that $p(\theta\,|E) \propto 1$) [5]. Again, the model with the smallest BIC value should be selected according to the BIC criterion. Note that the BIC score differs from the AIC score in that the penalty term also depends on the number of data points, T. When T is large enough, the BIC score penalizes model complexity more aggressively than the AIC score, which often proves better in practice. Also, the AICc (corrected AIC) criterion [28], which is a modified version of AIC, tends to work better than AIC for small sample sizes.

A **maximum a posteriori (MAP) estimate** of the joint configuration of structure and parameters is

$$(\hat{E}, \hat{\theta})_{MAPC} = \arg\max_{E,\theta} p(E, \theta \,|\, X; \beta, \gamma) = \arg\max_{E,\theta} p(X\,|E, \theta)\, p(E, \theta; \beta, \gamma)\,, \tag{2.65}$$

where the last equality follows from Equation 2.51 and recognizing that the normalization factor does not depend on E and θ. Here, the prior $p(E, \theta; \beta, \gamma)$ can also be thought of as a regularization term that can be used to penalize model complexity. For example, the prior on structure, $p(E; \beta)$, can be constructed to incur higher penalty on structures with higher number of edges in order to prevent overfitting. The structure component of the joint MAP estimate can alternatively be written as

$$\begin{aligned}\hat{E}_{MAPC} &= \arg\max_E \max_\theta p(X \mid E, \theta)\, p(E, \theta; \beta, \gamma) \\ &= \arg\max_E p(E; \beta) \max_\theta p(\theta; \gamma)\, p(X \mid E, \theta) \\ &= \arg\max_E p(E; \beta)\, p(\hat{\theta}_{MAP \mid E}; \gamma)\, p(X \mid E, \hat{\theta}_{MAP \mid E}), \end{aligned} \tag{2.66}$$

where $\hat{\theta}_{MAP \mid E}$ is the MAP estimate of parameters θ for a given structure E, i.e.,

$$\hat{\theta}_{MAP \mid E} = \arg\max_\theta p(\theta; \gamma)\, p(X \mid E, \theta). \tag{2.67}$$

Note that Equation 2.66 differs from Equation 2.58 only in the presence of prior, $p(E, \theta; \beta, \gamma)$. Therefore, an ML estimate of a structure can be thought of as a structure that belongs to the joint MAP estimate of the structure and parameters when their prior is "flat".

If we are only concerned about learning the structure (and treat parameters as nuisance variables), an alternative MAP estimate of a structure can be obtained by maximizing its marginal posterior distribution:

$$\hat{E}_{MAPM} = \arg\max_E p(E \mid X; \beta, \gamma) = \arg\max_E p(X \mid E; \gamma)\, p(E; \beta). \tag{2.68}$$

This can again be thought of as a model selection criterion, in which the model implied by structure E is evaluated by averaging data likelihood over all possible values of parameters for that structure, weighted by the prior probability of parameters (Equation 2.55), while the prior probability of structure serves as a model penalty. It is important to note however that Bayesian averaging over parameters (as in $p(X \mid E; \gamma)$) accounts for model complexity on its own. Namely, since the (weighted) average data likelihood is used instead of the maximum likelihood to evaluate a model, a more complex model is not necessarily better than a simpler one, even if the simpler model is contained within the complex model. Therefore, it is not necessarily the case that the prior on structure has to be used as a means of penalizing model complexity. For example, even if larger structures have higher prior probability, that may not necessarily result in overfitting.

Evaluatiing full posterior distribution

An alternative to structure point estimation is to compute the posterior distribution of all structures, as given by Equation 2.54, and then evaluate the probability of any

event of interest by a Bayesian averaging over structures. For example, the posterior probability of an edge $A \to B$ belonging to the interaction structure can be computed as

$$\begin{aligned} P(A \to B \mid X; \beta, \gamma) &= \sum_E \mathbb{I}(A \to B \in E)\, p(E \mid X; \beta, \gamma) \\ &= \sum_{E\,:\,A \to B \in E} p(E \mid X; \beta, \gamma)\,, \end{aligned} \tag{2.69}$$

where $\mathbb{I}$ is the *indicator function*.[3] This can be generalized to any event $\mathcal{A}$:

$$P(\mathcal{A} \mid X; \beta, \gamma) = \sum_E p(\mathcal{A} \mid E; \gamma)\, p(E \mid X; \beta, \gamma)\,. \tag{2.70}$$

If $\mathcal{A}(E)$ is an event that only depends on structure E, which we will call a **structural event**, then

$$\begin{aligned} P(\mathcal{A} \mid X; \beta, \gamma) &= \sum_E \mathbb{I}\left(\mathcal{A}(E)\right)\, p(E \mid X; \beta, \gamma) \\ &= \sum_{E\,:\,\mathcal{A}(E)} p(E \mid X; \beta, \gamma)\,. \end{aligned} \tag{2.71}$$

For instance, an event that an edge $A \to B$ belongs to the interaction structure, given in Equation 2.69 above, is an example of a structural event.

Similarly, the posterior probability of an edge $A \to C$ conditioned on the presence of edges $A \to B$ and $B \to C$ can be computed as

$$\begin{aligned} &P(A \to C \mid A \to B, B \to C, X; \beta, \gamma) \\ &\qquad = \frac{P(A \to C, A \to B, B \to C \mid X; \beta, \gamma)}{P(A \to B, B \to C \mid X; \beta, \gamma)} \\ &\qquad = \frac{\sum_{E\,:\,\{A \to C, A \to B, B \to C\} \in E} p(E \mid X; \beta, \gamma)}{\sum_{E\,:\,\{A \to B, B \to C\} \in E} p(E \mid X; \beta, \gamma)}\,, \end{aligned} \tag{2.72}$$

which can again be generalized to arbitrary events $\mathcal{A}$ and $\mathcal{B}$:

$$\begin{aligned} P(\mathcal{A} \mid \mathcal{B}, X; \beta, \gamma) &= \frac{P(\mathcal{A}, \mathcal{B} \mid X; \beta, \gamma)}{P(\mathcal{B} \mid X; \beta, \gamma)} \\ &= \frac{\sum_E p(\mathcal{A}, \mathcal{B} \mid E; \gamma)\, p(E \mid X; \beta, \gamma)}{\sum_E p(\mathcal{B} \mid E; \gamma)\, p(E \mid X; \beta, \gamma)}\,. \end{aligned} \tag{2.73}$$

[3] $\mathbb{I}(\mathit{cond}) = 1$ if *cond* is satisfied, 0 otherwise.

Finally, we define a **conditional structural event** $\mathcal{A}|\mathcal{B}(E)$ as an event that $\mathcal{A}(E)$ holds assuming that $\mathcal{B}(E)$ holds, whose probability can be computed as

$$P(\mathcal{A} \mid \mathcal{B}, X; \beta, \gamma) = \frac{\sum_{E:\mathcal{A}(E)\wedge\mathcal{B}(E)} p(E \mid X; \beta, \gamma)}{\sum_{E:\mathcal{B}(E)} p(E \mid X; \beta, \gamma)} . \tag{2.74}$$

An example of a conditional structural event is given in Equation 2.72.

Let us illustrate the variety of possible structural events with some more examples. The probability of an event that signal A has at most m parents, $indeg(A) = |pa(A)| \leq m$,[4] can be computed as

$$P(|pa(A)| \leq m \mid X; \beta, \gamma) = \sum_{E:|pa(A)|\leq m} p(E \mid X; \beta, \gamma) . \tag{2.75}$$

Another example is an event that signal A is a parent to at least m signals, $outdeg(A) = \sum_B \mathbb{I}(A \rightarrow B \in E) \geq m$.[5] The probability of this event can be computed as

$$P(outdeg(A) \geq m \mid X; \beta, \gamma) = \sum_{E:outdeg(A)\geq m} p(E \mid X; \beta, \gamma) . \tag{2.76}$$

Note that a point estimate of a structure only provides a prediction whether a structural event holds or not, and does not characterize the uncertainty of that estimate, which is, on the other hand, captured by evaluating the full posterior over structures. While this argument holds for any type of variable, and is the basis for using the Bayesian approach in the first place, it is particularly important in the case of structure inference. The number of possible structures is extremely large (superexponential in the number of nodes), with possibly many of them providing similarly good fit to the same data, even for relatively large data sizes. The uncertainty in the inferred structure is further increased in the cases when limited data is available and the data is imperfect (noisy and/or missing).

Note also that a frequentist approach can only be paired with maximum-likelihood point estimation, since it does not treat parameters (structure and parameters in this case) as probabilistic variables and therefore does not allow for computing their posterior distribution. It should be mentioned that various techniques for computing the confidence of estimated values or their statistical significance have been developed in the frequentist setting, but they have a different meaning – they only provide confidence in estimated values and therefore do not allow reasoning over different parameter values. For example, one may construct a hypothesis testing procedure that tests whether an edge $A \rightarrow B$ exists in the interaction structure. However, the result of such a procedure would be a conclusion whether the hypothesis should be accepted or not and an associated statistics that supports the decision (e.g., p-value). Furthermore, it may be the case that none of the hypothesis (presence or absence of an edge) is strongly supported

[4]The number of parents of a node is also called node in-degree.

[5]The number of children of a node is also called node out-degree.

by the data, but the associated statistics could not be used to compute the edge probability. Also, if hypotheses are made about the graph, it is very likely, due to the inherent uncertainty mentioned above, that none of the graphs would be "accepted". On the other hand, a Bayesian approach can be paired with maximum-a-posteriori point estimation, as well as as with evaluating the full posterior distribution, which characterizes the uncertainty in parameter values by computing their full posterior distribution, and thus is the approach of our choice in this thesis.

The examples above demonstrate the posterior analysis in the case of structural events, which are binary functions of the structure. The same type of analysis can be performed for any type of function by evaluating the posterior probability[6] of each possible outcome. For example, the posterior distribution of a node out-degree can be computed as

$$P(outdeg(A) = m \mid X; \beta, \gamma) = \sum_{E\,:\,outdeg(A)=m} p(E \mid X; \beta, \gamma)\,. \tag{2.77}$$

More generally, if $f(E)$ is an arbitrary function of the structure E, which we will refer to as a **structural property**, the posterior probability that it takes a particular value v can be computed as

$$P(f(E) = v \mid X; \beta, \gamma) = \sum_{E\,:\,f(E)=v} p(E \mid X; \beta, \gamma)\,. \tag{2.78}$$

Note that structural events are a special case of structural properties – binary properties. It is worth noting that binary properties exhibit the weakness of point estimation the most. If the point estimation of such property is wrong, it misleads further analysis. On the other hand, estimates of some properties can be useful even if they are wrong. For example a point estimate of a property whose value lives in an "ordered" space, such as node out-degree, provides insight into which area of the space its value may belong to (e.g., whether the out-degree of a node is high or low). Still, even in such cases, the Bayesian approach provides more information about such a property by evaluating its entire posterior distribution.

■ 2.6.3 Complexity of Bayesian network structure inference

Bayesian network structure learning is a hard problem – NP-complete in general [10]. First of all, the number of possible static Bayesian networks with N nodes is huge. It is the same as the number of directed acyclic graphs (DAGs) with N nodes, which we denote as g_N. It can be shown that g_N is superexonential in N with exponent $\Theta(N^2)$.

Lemma 2.6.1. $g_N \geq 2^{\binom{N}{2}}$.

Proof. Let $\pi = (i_1, i_2, \ldots, i_N)$ be an arbitrary permutation of node indices $1, 2, \ldots, N$. Let us consider only directed graphs in which each edge $(i_j \to i_k)$ must satisfy $j < k$

[6] or density in the case of continuous-valued functions

(i.e., edges must point from left to right with respect to the permutation). We will say that these graphs "respect" the permutation π, or that π is a topological order for them.[7] They can be constructed by choosing independently for each pair of indices $j < k$ whether to have an edge $i_j \to i_k$ or not. There are $\binom{N}{2}$ such pairs. Therefore, the number of graphs that respect the permutation π is $g_\pi = 2^{\binom{N}{2}}$, for any permutation of N nodes. Such graphs are DAGs, since any cycle would have to contain at least one edge going from right to left in the permutation, from which it follows that $g_N \geq g_\pi$. ■

Lemma 2.6.2. $g_N \leq 3^{\binom{N}{2}}$.

Proof. Let us consider all directed graphs with N nodes that do not contain cycles of length 1 (self-loops) nor cycles of length 2 (pairs of edges $i \to j$ and $j \to i$, for any i and j). They can be constructed by choosing independently for each pair of nodes $i \neq j$ whether there is an edge $i \to j$, an edge $j \to i$, or no edge between them. Therefore, there are $3^{\binom{N}{2}}$ such graphs. These graphs necessarily include all DAGs, from which the statement of the Lemma follows. ■

Theorem 2.6.1. $g_N = 2^{\Theta(N^2)}$.

Proof. From Lemmas 2.6.1 and 2.6.2

$$\frac{N(N-1)}{2} \leq \log_2(g_N) \leq \log_2(3)\frac{N(N-1)}{2},$$

from which the statement of the theorem follows. ■

From the proof of Lemma 2.6.1, one may attempt to conclude that $g_N = N!\,2^{\binom{N}{2}}$, as there are $N!$ possible permutations of nodes and $2^{\binom{N}{2}}$ possible DAGs that respect each permutation. This is however not true because some DAGs respect more than one permutation and are therefore counted more than once. For example, a graph with no edges is a DAG that respects all permutations. On the other hand, since $N!\,2^{\binom{N}{2}}$ is an overestimate of the number of DAGs, it can serve as an upper bound. In fact, it is asymptotically a tighter upper bound than $3^{\binom{N}{2}}$ from Lemma 2.6.2. To see that, note that $\log N! = \sum_{i=1}^{N} \log i = \Theta(N \log N) = o(N^2)$,[8] from which it follows that $\log N!\,2^{\binom{N}{2}} = o(N^2) + \frac{N(N-1)}{2}$, which is clearly smaller than $3^{\binom{N}{2}} = \log_2(3)\frac{N(N-1)}{2}$ for large enough N. The exact number of DAGs with N nodes can be computed recursively due to Robinson [47] as

$$g_N = \sum_{m=1}^{N} (-1)^{m-1} \binom{N}{m} 2^{m(N-m)} g_{N-m}, \tag{2.79}$$

starting with $g_0 = 1$ (there is only one DAG with 0 nodes – empty graph).

[7] Note that each graph can have multiple topological orders.

[8] $\sum_{i=1}^{N} \log i = \Theta(N \log N)$ follows simply from $\frac{N}{2} \log \frac{N}{2} \leq \sum_{i=1}^{N} \log i \leq N \log N$.

Theorem 2.6.1 shows that the number of possible static Bayesian networks is superexponential in the number of nodes. Therefore, the complexity of evaluating the full posterior over such networks is also superexponential, since the posterior probabilty of each possible structure must be evaluated. It does not immediately follow that structure point estimation incurs the same complexity (many optimization problems with exponential number of possible solutions are solvable in polynomial time by exploring some structure in the solution space – e.g., Dijskstra's algorithm for finding the shortest paths in a graph). However, Chickering [10] has shown that finding the "best" structure is NP-complete under very general assumptions – existence of a structure scoring function (e.g., marginal data likelihood given structure as in Equation 2.55) and structure penalty function (e.g., AIC/BIC penalty or structure prior probability). Hence, both ML and MAP structure estimation (as well as any other "reasonable" point estimation method) are NP-complete problems.

Learning a homogenous dynamic Bayesian network is a very similar problem. The number of possible such networks with N signals, d_N, is also superexponential in the number of signals.

Theorem 2.6.2. $d_N = 2^{N^2}$.

Proof. All homogenous dynamic Bayesian networks with N signals, $X^1, X^2, \ldots, X^N$ are fully determined by the edges between variables at any two neighboring time points $t-1$ and t. For each pair of variables X_{t-1}^i and X_t^j, there are two choices: there is no edge between them or there is an edge $X_{t-1}^i \rightarrow X_t^j$. There are N^2 such pairs, and each choice for an edge can be made independently. Therefore, there are 2^{N^2} possible structures. Note that this is exactly the number of bipartite graphs between two sets of nodes of size N. ■

Consequently, d_N is also the number of possible interaction graphs between N signals at any given time. Note that calculating this number is simpler than calculating the number of static Bayesian networks. The main reason is that dynamic Bayesian networks implictly assume ordering of nodes (which is temporal ordering) and thus do not involve permutation selection. As a consequence, each edge can be chosen independently. The same does not hold for static networks – choosing a subset of edges may prevent choosing some other edges in order to satisfy graph acyclicity (edges can only be chosen independently when conditioned on a particular permutation).

Obviously, the complexity of the Bayesian inference over homogenous DBNs is also superexponential, which is of our primary concern. On the other hand, to the best of our knowledge, it is not clear from the existing literature whether the result of Chickering holds in this case as well. In other words, it is not cleat whether homogenous DBN structure point estimation is necessarily NP-complete under the same assumptions and whether there are some specific instances of that problem in which the simplified structure of the solution space (no permutations involved) can be exploited to obtain polynomial-time algorithms. While this is certainly a very important and interesting problem, it will not be of a concern in this thesis, as we are primarily interested in the

# nodes	# static BNs	# interactions
1	1	2
2	3	16
3	25	512
4	543	65,536
5	29,281	33,554,432
6	3,781,503	$6.87 * 10^{10}$
7	$1.14 * 10^{09}$	$5.63 * 10^{14}$
8	$7.84 * 10^{11}$	$1.84 * 10^{19}$
9	$1.21 * 10^{15}$	$2.42 * 10^{24}$
10	$4.18 * 10^{18}$	$1.27 * 10^{30}$

Table 2.1: The number of possible static Bayesian networks and homogenous interaction structures as a function of the number of nodes.

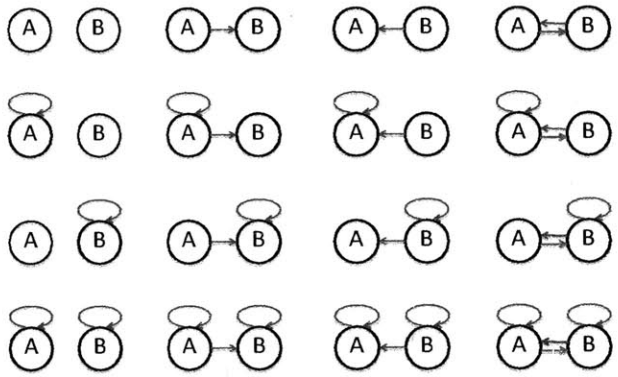


Figure 2.6: There are 16 possible interaction structures among 2 signals.

Bayesian approach.

■ 2.6.4 Prior for efficient structure inference

Exact Bayesian inference over both static Bayesian networks and homogenous DBNs (or, equivalently, homogenous interaction structures) is computationally tractable only when the number of nodes or signals is very small. The number of possible structures for both types of networks as a function of the number of nodes/signals is shown in Table 2.1. For example, even for only 2 nodes there are already 16 possible interaction structures, which are shown in Figure 2.6.

In order to allow for tractable inference over structure, an approximate algorithm must be employed or some assumptions must be made in order to reduce the space of allowed structures (or both). The most widely used class of algorithms for approximate Bayesian inference over structures are sampling algorithms. For example, Markov chain Monte Carlo (MCMC) simulation has been used to generate samples from the poste-

rior distribution over structures [19, 36], which can be used to estimate the posterior probability of any structural property:

$$P(f(E) = v \mid X; \beta, \gamma) \approx \frac{1}{N_s} \sum_{i=1}^{N_s} \mathbb{I}\left(f(\hat{E}_i) = v\right) , \tag{2.80}$$

where $\hat{E}_1, \ldots, \hat{E}_{N_s} \sim P(E \mid X; \beta, \gamma)$ are N_s samples from the structure posterior. The key challenge in such approaches is to construct a proposal distribution that leads to efficient exploration of the space of structures with respect to their posterior probability.

We choose the latter approach to tractable inference over structures, which is to impose constraints that reduce the space of possible structures and perform exact inference over the remaining structures. We follow the work of Siracusa and Fisher [50] and use the following two assumptions: (1) modular prior assumption, which allows independent reasoning over parent sets of signals and reduces the complexity of inference to exponential, and (2) additional constraints on parent sets, such as bounded in-degree assumption, which further reduce the complexity of inference to polynomial in the number of signals.

A **modular prior** on structure and parameters [7, 11, 19, 27] is based on the following assumptions:

- $p(E; \beta) = \prod_{i=1}^{N} p(pa(i); \beta)$ (structure modularity)
- $p(\theta | E; \gamma) = \prod_{i=1}^{N} p(\theta^i | E; \gamma)$ (global parameter independence)
- $p(\theta^i | E; \gamma) = p(\theta^i | pa(i); \gamma)$ (param. modularity).

The "structure modularity" assumption states that the parent sets of different signals are independent of each other with respect to the prior probability of structure. The "global parameter independence" assumption states that the parameters of the dependence models of different signals are independent of each other with respect to their prior probability given structure. Finally, the "parameter modularity" assumption states that the prior probability of parameters of the dependence model of a signal only depends on the parent set of that signal, and is therefore independent of the parent sets of other signals. These three assumptions can thus be written in the form of prior independence statements:

- $pa(i) \perp\!\!\!\perp pa(j)$, $\forall i, j : i \neq j$ (structure modularity)
- $\theta^i \perp\!\!\!\perp \theta^j \mid E$, $\forall i, j : i \neq j$ (global parameter independence)
- $\theta^i \perp\!\!\!\perp pa(j)$, $\forall i, j : i \neq j$ (param. modularity),

as well as summarized in the following statement: The modular prior on structure and parameters decomposes as a product of priors on parent sets of individual signals and associated parameters,

$$p(E, \theta; \beta, \gamma) = \prod_{i=1}^{N} p(pa(i); \beta)\, p(\theta^i \mid pa(i); \gamma) . \tag{2.81}$$

Modular prior on structure and parameters can be constructed in the following way. There is a separate prior distribution for a parent set of each node i, which has a general form

$$p(pa(i);\beta) = \frac{1}{B_i}\beta_{i,pa(i)} \propto \beta_{i,pa(i)}\,, \tag{2.82}$$

where $B_i = \sum_s \beta_{i,s}$ are normalization constants. β is no longer a collection of parameters per structure (as in Equation 2.50), but rather a collection of parameters $\{\beta_{i,pa(i)}\}$ (one parameter for each possible parent set of each signal). Similarly, γ is a collection of hyperparameters $\{\gamma_{i,pa(i)}\}$, such that $p(\theta^i|pa(i);\gamma) = p(\theta^i;\gamma_{i,pa(i)})$.

Modularity is also reflected in the posterior:

$$p(E,\theta\,|X;\beta,\gamma) = \prod_{i=1}^{N} p(pa(i)\,|\,X;\beta)\; p(\theta^i\,|X,pa(i);\gamma)\;. \tag{2.83}$$

For static Bayesian networks, the modular prior assumptions are meaningful when the ordering (permutation) of nodes is fixed. The reason for that is that, in general, parent sets of nodes cannot be chosen independently as that may result in creating a cycle, which is a global relationship. However, when the ordering of nodes is fixed, and, for each node, only the parent sets that respect that ordering are allowed (i.e., only parents that are to the left of a node with respect to the permutation), then parent sets of nodes can indeed be chosen independently, as any combination of choices would result in a structure that respects the permutation. This property was first exploited by Buntine [7] and Cooper and Dietterich [11], which assume that the order of variables is known (e.g., determined by a domain expert), while Friedman and Koller [19] combine it with a procedure for sampling node permutations from their posterior distribution.

On the other hand, the modular prior assumptions can be applied unconditionally (i.e., without any further assumption) in the case of interaction graphs. This follows simply from the fact that interaction graphs do not need to be acyclic (i.e., any directed graph is permitted) and parent sets can be chosen independently for each signal [50].

As a result, parent sets can be chosen independently for each signal [50], and the total number of parent sets to consider is $N2^N$, which is exponential in the number of signals.

If, in addition, the number of parents of each signal is bounded by some constant M (a structure with bounded in-degree [11, 19, 27]), the number of parent sets to evaluate is further reduced to $O(N^{M+1})$, which is polynomial in N.

■ 2.6.5 Related Work

Learning Bayesian network structure (under reasonable assumptions) is NP hard [10]. However, there has been an extensive body of work on exact and approximate methods. While some work employs direct causality testing (constraint-based methods) [53, 57], most researchers focus on a Bayesian approach, or a score-based approach that possibly has a Bayesian interpretation. A number of heuristic methods for finding a structure

with the maximum a posteriori (MAP) probability have been developed [7, 11, 27]. A few assumptions are typically introduced to reduce the search space. Assuming a known ordering of variables (e.g., [7, 11]), for which edges are always directed from "left" to "right", eliminates a global constraint of the structure being acyclic – namely, a parent set of each node can be chosen independently and the graph will still be acyclic. If, in addition, a prior on the structure and parameters is modular (e.g., [7, 11, 19, 27]), inference over each node's parent set can be performed independently, and the complexity of structure inference is reduced from superexponential to exponential in the number of nodes. Introducing a bound on the number of parents of each node further reduces the complexity to polynomial (e.g., [11, 19, 27]).

Learning the best structure from limited data is challenging. There may be many structures that are similarly "good". Also, the probability of learning the correct structure decreases rapidly with the number of objects. Therefore, for all but small problems, a large amount of data is needed to avoid errors. In addition, there are typically multiple structures that encode the same set of independences among involved variables (Markov equivalence class), leading to identifiability issues. On the other hand, in most cases, the structure itself is not of direct interest, but rather some of its properties. For example, is there an edge between two nodes? Instead of reading these properties from a potentially incorrect single learned structure, it is possible to compute their posterior probabilities via Bayesian structure averaging, as suggested by Cooper and Herskovits [11]. This approach does not provide definite answers. However, it fully characterizes uncertainty in the structure and any of its properties. This additional information is especially valuable when decisions that follow the analysis are postponed to a further analysis (e.g., by a domain expert). Note that another important goal of structure learning is to obtain better predictive models. It has been shown that Bayesian averaging improves predictive performance over inference based on a single model (e.g., [36]).

The powerful methodology of MCMC [58] was first used for Bayesian structure averaging by Madigan et al. [36]. However, this method tends to mix poorly and does not explore well the space of structures, due to the local nature of MCMC moves (at most one edge is added or removed from the graph in a single step). Friedman and Koller [19] developed a method that combines MCMC sampling over orders of variables with exact inference over structures for a given order (which is polynomial in the number of nodes by the assumption of modular prior and bounded number of parents). The space of orders is much smoother in the posterior over structures than the whole space of structures, leading to a significantly better performance of the MCMC method. Niinimäki et al. [39] further improve MCMC performance by sampling over even smoother space of partial orders. One drawback of the methods that sample from linear or partial orders of variables is their inability to explicitly specify priors on structures. Grzegorczyk and Husmeier [25] improved the original MCMC algorithm over structures (DAGs) by introducing a new edge reversal proposal move.

Similarly, learning DBN's has been addressed by Friedman et al. [20]. The number

of structures to consider can again be reduced by imposing constraints, for example, by bounding the number of allowed parents per object (Friedman and Koller [19], Siracusa and Fisher [50]). For some special classes of structures, such as trees, it is possible to reason over marginal events efficiently without explicit enumeration (Meila and Jaakkola [37], Siracusa and Fisher [50]).

■ 2.7 Bayesian Learning of Switching Dependence Structure

In order to learn time-varying interaction from time-series data, Siracusa and Fisher [49, 50] assume that the dependence model switches over time between K distinct models, $\tilde{\mathcal{M}}_k = (\tilde{E}_k, \tilde{\theta}_k), k = 1, \ldots, K$. More formally, for each time point t, $\mathcal{M}_t = \tilde{\mathcal{M}}_k$ for some k, $1 \leq k \leq K$. One interaction may be active for some period of time, followed by a different interaction over another period of time, and so on, switching between a pool of possible interactions. This is illustrated in Figure 1.1. Let Z_t, $1 \leq t \leq T$, be a discrete random variable that represents an index of a dependence model active at time point t; i.e., $\mathcal{M}_t = \tilde{\mathcal{M}}_{Z_t}$, $Z_t \in \{1, \ldots, K\}$. Equation 2.46 can now be rewritten as

$$\begin{aligned} p(X_t|X_{t-1}, Z_t, \tilde{E}, \tilde{\theta}) &= p(X_t|X_{t-1}, \tilde{E}_{Z_t}, \tilde{\theta}_{Z_t}) \\ &= \prod_{i=1}^{N} p(X_t^i|X_{t-1}^{\tilde{pa}(i,Z_t)}, \tilde{\theta}_{Z_t}^i) \,, \end{aligned} \tag{2.84}$$

where $(\tilde{E}, \tilde{\theta}) = \{(\tilde{E}_k, \tilde{\theta}_k)\}_{k=1}^K$ is a collection of all K models and $\tilde{pa}(i,k)$ is a parent set of signal i in $\tilde{E}_k$, and Equation 2.47 as

$$p(X|Z, \tilde{E}, \tilde{\theta}) = p(X_0|\theta_0) \prod_{t=1}^{T} p(X_t|X_{t-1}, Z_t, \tilde{E}, \tilde{\theta}) \,, \tag{2.85}$$

where $Z = \{Z_t\}_{t=1}^T$. To distinguish from signal state, we call Z_t a switching state (at time t) and Z a switching sequence. Furthermore, it is assumed that Z forms a first order Markov chain:

$$p(Z) = p(Z_1) \prod_{t=2}^{T} p(Z_t|Z_{t-1}) = \pi_{Z_1} \prod_{t=2}^{T} \pi_{Z_{t-1}, Z_t} \,, \tag{2.86}$$

where $\pi_{i,j}$ is a transition probability from state i to state j and π_i is the initial probability of state i.

The full STIM generative model, shown in Figure 2.7, incorporates probabilistic models described above along with priors on structures and parameters:

- Multinomials π are sampled from Dirichlet priors parametrized by α as
  $(\pi_1, \ldots, \pi_K) \sim \mathcal{D}ir(\alpha_1, \ldots, \alpha_K)$,
  $(\pi_{i,1}, \ldots, \pi_{i,K}) \sim \mathcal{D}ir(\alpha_{i,1}, \ldots, \alpha_{i,K})$ $\forall i$.

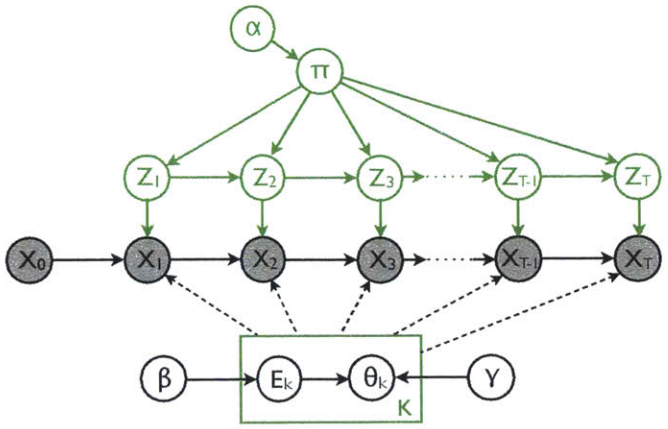


Figure 2.7: Switching temporal interaction model of Siracusa and Fisher [50].

- K structures $\tilde{E}_k$ and parameters $\tilde{\theta}_k$ are sampled from the corresponding priors as $\tilde{E}_k \sim p(E;\beta)$, $\tilde{\theta}_k \sim p(\theta|\tilde{E}_k;\gamma)$, $\forall k$.
- Initial value X_0 is generated as $X_0 \sim p(X_0|\theta_0)$.
- For each $t = 1, 2, \ldots, T$ (in that order), values of Z_t and X_t are sampled as
  $Z_t \sim \mathcal{M}ult(\pi_{Z_{t-1},1}, \ldots, \pi_{Z_{t-1},K})$ or
  $Z_t \sim \mathcal{M}ult(\pi_1, \ldots, \pi_K)$ if $t = 1$,
  $X_t \sim p(X_t|X_{t-1}, \tilde{E}_{Z_t}, \tilde{\theta}_{Z_t})$.

Algorithm 2.1 STIM Gibbs sampler

1. $Z \sim p(Z|X, \tilde{E}, \tilde{\theta}, \pi)$
2. $\pi \sim p(\pi|Z;\alpha)$
3. $\tilde{E}, \tilde{\theta} \sim p(\tilde{E}, \tilde{\theta}|Z, X;\beta,\gamma)$

Inference in the STIM is done using a Gibbs sampling procedure shown in Algorithm 2.1. Sampling of the K dependence models (structures and parameters) is done as if each of these models is homogenous. Namely, since this step the switching sequence Z is conditioned on in this step and is therefore assumed know, pairs (X_{t-1}, X_t) pertained to each state are pulled together to perform an update on that states' model. This procedure is shown in Algorithm 2.2. Note that this step is efficient when a modular bounded-indegreee prior on structures is used in conjunction with a conjugate prior on dependence model parameters (Algorithm 2.3). In case of a linear Gaussian dependence model with a matrix-normal inverse-Wishart prior, the procedure is shown in Algorithm 2.4. The procedure for sampling parameters π of multinomials given the switching sequence Z (step 2) is straightforward as the Dirichlet distribution is conjugate to the

multinomial, and is shown in Algorithm 2.5. Given the state sequence X and the dependence models $\{\tilde{E}_k, \tilde{\theta}_k\}_{k=1}^K$, a sample of a switching sequence (step 1) is generated via a backward message-passing forward sampling algorithm, which we now discuss in more detail.

Algorithm 2.2 Sampling structures and parameters of the K dependence models: $\tilde{E}, \tilde{\theta} \sim p(\tilde{E}, \tilde{\theta} | Z, X; \beta, \gamma)$

$$
\begin{aligned}
&for\ k = 1, \ldots, K \\
&\quad for\ E \in \mathcal{E}_k \\
&\qquad \beta'^k_E = \beta^k_E\, P\left(\{X_t\}_{t:Z_t=k} \,|\, \{X_{t-1}\}_{t:Z_t=k}, E; \gamma^k_E\right) \\
&\quad \tilde{E}_k \sim Categorical\left(\{\beta'^k_E\}_{E\in\mathcal{E}_k}\right) \quad // P(\tilde{E}_k = E) \propto \beta'^k_E \\
&\quad \tilde{\theta}_k \sim P\left(\tilde{\theta}_k \,|\, \{X_t, X_{t-1}\}_{t:Z_t=k}, \tilde{E}_k, \gamma^k_{\tilde{E}_k}\right)
\end{aligned}
$$

Algorithm 2.3 Sampling structures and parameters of the K dependence models with modular prior: $\tilde{E}, \tilde{\theta} \sim p(\tilde{E}, \tilde{\theta} | Z, X; \beta, \gamma)$

$$
\begin{aligned}
&for\ k = 1, \ldots, K \\
&\quad for\ i = 1, \ldots, N \\
&\qquad for\ s \in \mathcal{PA}^i_k \\
&\qquad\quad \beta'^k_{i,s} = \beta^k_{i,s}\, P\left(\{X^i_t\}_{t:Z_t=k} \,|\, \{X^s_{t-1}\}_{t:Z_t=k}, s; \gamma^k_{i,s}\right) \\
&\qquad \tilde{pa}(i,k) \sim Categorical\left(\{\beta'^k_{i,s}\}_{s\in\mathcal{PA}_k}\right) \quad // P(\tilde{pa}(i,k) = s) \propto \beta'^k_{i,s} \\
&\qquad \tilde{\theta}^i_k \sim P\left(\tilde{\theta}^i_k \,|\, \{X^i_t, X^{\tilde{pa}(i,k)}_{t-1}\}_{t:Z_t=k}, \tilde{pa}(i,k), \gamma^k_{i,\tilde{pa}(i,k)}\right)
\end{aligned}
$$

■ 2.7.1 Batch sampling of the switching state sequence (step 1)

A conditional distribution of Z can be decomposed as

$$
P(Z|X, \tilde{E}, \tilde{\theta}, \pi) = P(Z_1|X, \tilde{E}, \tilde{\theta}, \pi) \prod_{t=2}^{T} P(Z_t|Z_{1:t-1}, X, \tilde{E}, \tilde{\theta}, \pi)\,. \tag{2.87}
$$

Algorithm 2.4 Sampling structures and parameters of the K dependence models with modular prior in LG-SSIM: $\tilde{E}, \tilde{\theta} \sim p(\tilde{E}, \tilde{\theta} | Z, X; \beta, \gamma)$

$$
\begin{aligned}
&for\ k = 1, \ldots, K \\
&\quad for\ i = 1, \ldots, N \\
&\quad\quad for\ s \in \mathcal{PA}_k^i \\
&\quad\quad\quad \Omega'^k_{i,s} = \left({\Omega^k_{i,s}}^{-1} + \sum_{t:Z_t=k} X^s_{t-1} {X^s_{t-1}}^T \right)^{-1} \\
&\quad\quad\quad M'^k_{i,s} = \left(M^k_{i,s} {\Omega^k_{i,s}}^{-1} + \sum_{t:Z_t=k} X^i_t {X^s_{t-1}}^T \right) \Omega'^k_{i,s} \\
&\quad\quad\quad \kappa'^k_{i,s} = \kappa^k_{i,s} + |\{t : Z_t = k\}| \\
&\quad\quad\quad \Psi'^k_{i,s} = \Psi^k_{i,s} + \sum_{t:Z_t=k} X^i_t {X^i_t}^T + M^k_{i,s} {\Omega^k_{i,s}}^{-1} {M^k_{i,s}}^T + M'^k_{i,s} {\Omega'^k_{i,s}}^{-1} {M'^k_{i,s}}^T \\
&\quad\quad\quad P\left(\{X^i_t\}_{t:Z_t=k} \,|\, \{X^s_{t-1}\}_{t:Z_t=k}, s; \gamma^k_{i,s} \right) = \frac{|\Omega'^k_{i,s}|^{d/2} |\Psi^k_{i,s}|^{\kappa^k_{i,s}/2} \Gamma_d(\frac{\kappa'}{2})}{|\Omega^k_{i,s}|^{d/2} |\Psi'^k_{i,s}|^{\kappa'^k_{i,s}/2} \Gamma_d(\frac{\kappa}{2}) \pi^{\frac{Td}{2}}} \\
&\quad\quad\quad \beta'^k_{i,s} = \beta^k_{i,s}\, P\left(\{X^i_t\}_{t:Z_t=k} \,|\, \{X^s_{t-1}\}_{t:Z_t=k}, s; \gamma^k_{i,s} \right) \\
&\quad \tilde{pa}(i,k) \sim Categorical\left(\{\beta'^k_{i,s}\}_{s \in \mathcal{PA}_k} \right) \quad // \, P(\tilde{pa}(i,k) = s) \propto \beta'^k_{i,s} \\
&\quad (\tilde{A}^i_k, \tilde{Q}^i_k) \sim \mathcal{MN}\text{-}\mathcal{IW}\left(\tilde{A}^i_k, \tilde{Q}^i_k; M'^k_{i,\tilde{pa}(i,k)}, \Omega'^k_{i,\tilde{pa}(i,k)}, \Psi'^k_{i,\tilde{pa}(i,k)}, \kappa'^k_{i,\tilde{pa}(i,k)} \right)
\end{aligned}
$$

Algorithm 2.5 Sampling of the switching sequence multinomials: $\pi \sim p(\pi | Z; \alpha)$

Dirichlet priors conjugate update

$$
\begin{aligned}
&\alpha' = \alpha \\
&\alpha'_{Z_1} = \alpha'_{Z_1} + 1 \\
&for\ t = 2, \ldots, T \\
&\quad \alpha'_{Z_{t-1}, Z_t} = \alpha'_{Z_{t-1}, Z_t} + 1
\end{aligned}
$$

Sampling multinomials

$$
\begin{aligned}
&(\pi_1, \ldots, \pi_K) \sim \mathcal{D}ir(\alpha'_1, \ldots, \alpha'_K) \\
&for\ i = 1, \ldots, K \\
&\quad (\pi_{i,1}, \ldots, \pi_{i,K}) \sim \mathcal{D}ir(\alpha'_{i,1}, \ldots, \alpha'_{i,K})
\end{aligned}
$$

Therefore, the following forward sampling procedure:

$$
\begin{aligned}
Z_1 &\sim P(Z_1|X, \tilde{E}, \tilde{\theta}, \pi) \\
Z_2 &\sim P(Z_2|Z_1, X, \tilde{E}, \tilde{\theta}, \pi) \\
&\cdots \\
Z_T &\sim P(Z_T|Z_{1:T-1}, X, \tilde{E}, \tilde{\theta}, \pi)
\end{aligned} \tag{2.88}
$$

generates a joint sample of variables $Z_1, \ldots, Z_T$ from the above conditional distribution. Here, "forward" refers to the temporal order in which switching variables are sampled. $P(Z_1|X, \tilde{E}, \tilde{\theta}, \pi)$ can be computed in the following way:

$$
\begin{aligned}
&P(Z_1|X, \tilde{E}, \tilde{\theta}, \pi) \\
&\propto P(Z_1, X|\tilde{E}, \tilde{\theta}, \pi) \\
&= \sum_{Z_{2:T}} P(Z_1, Z_{2:T}, X|\tilde{E}, \tilde{\theta}, \pi) \\
&= \sum_{Z_{2:T}} P(X_0) \prod_{t=1}^{T} P(Z_t|Z_{t-1}, \pi) P(X_t|X_{t-1}, \tilde{E}_{Z_t}, \tilde{\theta}_{Z_t}) \\
&\propto \sum_{Z_{2:T-1}} \left[\prod_{t=1}^{T-1} P(Z_t|Z_{t-1}, \pi) P(X_t|X_{t-1}, \tilde{E}_{Z_t}, \tilde{\theta}_{Z_t}) \right] \\
&\qquad \times \underbrace{\sum_{Z_T} P(Z_T|Z_{T-1}, \pi) P(X_T|X_{T-1}, \tilde{E}_{Z_T}, \tilde{\theta}_{Z_T})}_{m^{T-1}(Z_{T-1})} \\
&= \sum_{Z_{2:T-2}} \left[\prod_{t=1}^{T-2} P(Z_t|Z_{t-1}, \pi) P(X_t|X_{t-1}, \tilde{E}_{Z_t}, \tilde{\theta}_{Z_t}) \right] \\
&\qquad \times \underbrace{\sum_{Z_{T-1}} P(Z_{T-1}|Z_{T-2}, \pi) P(X_{T-1}|X_{T-2}, \tilde{E}_{Z_{T-1}}, \tilde{\theta}_{Z_{T-1}}) \, m^{T-1}(Z_{T-1})}_{m^{T-2}(Z_{T-2})} \\
&\qquad \vdots \\
&= P(Z_1|\pi) P(X_1|X_0, \tilde{E}_{Z_1}, \tilde{\theta}_{Z_1}) \underbrace{\sum_{Z_2} P(Z_2|Z_1, \pi) P(X_2|X_1, \tilde{E}_{Z_2}, \tilde{\theta}_{Z_2}) \, m^2(Z_2)}_{m^1(Z_1)} \\
&= P(Z_1|\pi) P(X_1|X_0, \tilde{E}_{Z_1}, \tilde{\theta}_{Z_1}) \, m^1(Z_1) ,
\end{aligned} \tag{2.89}
$$

where $P(Z_1|Z_0) \equiv P(Z_1)$ for convenience, and messages are defined recursively as

$$m^T(z) = 1, \quad \forall z = 1, \ldots, K$$
$$m^t(z) = \sum_{Z_{t+1}} P(Z_{t+1}|z, \pi) P(X_{t+1}|X_t, \tilde{E}_{Z_{t+1}}, \tilde{\theta}_{Z_{t+1}})\, m^{t+1}(Z_{t+1}),$$
$$\forall z = 1, \ldots, K, \ \forall t = 1, \ldots, T-1. \tag{2.90}$$

Note that the message $m^T(z) = 1$ is introduced for initialization convenience. It represents a uniform distribution, which can be interpreted as that no information about time point $T+1$ is "coming into" time point T. Messages can also be written in a non-recursive form as

$$m^t(z) = \sum_{Z_{t+1:T}} P(Z_{t+1}|z, \pi) \prod_{i=t+2}^{T} P(Z_i|Z_{i-1}, \pi) \prod_{i=t+1}^{T} P(X_i|X_{i-1}, \tilde{E}_{Z_i}, \tilde{\theta}_{Z_i}),$$
$$\forall z = 1, \ldots, K, \ \forall t = 1, \ldots, T-1. \tag{2.91}$$

Finally, $P(Z_t|Z_{1:t-1}, X, \tilde{E}, \tilde{\theta}, \pi)$, for each $t = 2, \ldots, T$, can be computed as:

$$\begin{aligned}
&P(Z_t|Z_{1:t-1}, X, \tilde{E}, \tilde{\theta}, \pi) \\
&\propto P(Z_t, X|Z_{1:t-1}, \tilde{E}, \tilde{\theta}, \pi) \\
&= \sum_{Z_{t+1:T}} P(Z_t, Z_{t+1:T}, X|Z_{1:t-1}, \tilde{E}, \tilde{\theta}, \pi) \\
&\propto \sum_{Z_{t+1:T}} P(Z_t, Z_{t+1:T}, X_{t:T}|Z_{t-1}, X_{t-1}, \tilde{E}, \tilde{\theta}, \pi) \\
&= P(Z_t|Z_{t-1}, \pi) P(X_t|X_{t-1}, \tilde{E}_{Z_t}, \tilde{\theta}_{Z_t}) \sum_{Z_{t+1:T}} \prod_{i=t+1}^{T} P(Z_i|Z_{i-1}, \pi) P(X_i|X_{i-1}, \tilde{E}_{Z_i}, \tilde{\theta}_{Z_i}) \\
&= P(Z_t|Z_{t-1}, \pi) P(X_t|X_{t-1}, \tilde{E}_{Z_t}, \tilde{\theta}_{Z_t})\, m^t(Z_t).
\end{aligned} \tag{2.92}$$

Observe that the messages, previosly computed in a backward fashion, are reused to shortcut the computation of each $P(Z_t|Z_{1:t-1}, X, \tilde{E}, \tilde{\theta}, \pi)$, which are computed in a forward fashion.

The full procedure is given in Algoritm 2.6. Evaluating $P(X_t|X_{t-1}, \tilde{E}_{Z_t}, \tilde{\theta}_{Z_t})$, $t = 1, \ldots, T$, $Z_t = 1, \ldots, K$, requires $O(TKN)$ time in total.[9] Computing all messages recursively takes $O(TK^2)$ time (computing each $m^t(z)$ for $t < T$ requires a summation over K values). Finally, once the messages are computed, forward sampling of sequence Z requires $O(TK)$ time. Therefore, the total time needed for sampling Z is $O(TK(K+N))$.

[9]Recall that X_t is a collection of variables of N signals.

Algorithm 2.6 Batch sampling of the switching state sequence: $Z \sim p(Z|X, \tilde{E}, \tilde{\theta}, \pi)$

Backward message passing

$m^T(z) = 1, \quad \forall z = 1, \ldots, K$

for $t = T-1, \ldots, 1$

$$m^t(z) = \sum_{Z_{t+1}} P(Z_{t+1}|z) P(X_{t+1}|X_t, \tilde{E}_{Z_{t+1}}, \tilde{\theta}_{Z_{t+1}})\, m^{t+1}(Z_{t+1}), \quad \forall z = 1, \ldots, K$$

Forward sampling

$P(Z_1|X, \tilde{E}, \tilde{\theta}) \propto P(Z_1) P(X_1|X_0, \tilde{E}_{Z_1}, \tilde{\theta}_{Z_1})\, m^1(Z_1)$

$Z_1 \sim P(Z_1|X, \tilde{E}, \tilde{\theta})$

for $t = 2, \ldots, T$

$P(Z_t|Z_{1:t-1}, X, \tilde{E}, \tilde{\theta}) \propto P(Z_t|Z_{t-1}) P(X_t|X_{t-1}, \tilde{E}_{Z_t}, \tilde{\theta}_{Z_t})\, m^t(Z_t)$

$Z_t \sim P(Z_t|Z_{1:t-1}, X, \tilde{E}, \tilde{\theta})$

Chapter 3

SSIM: State-Space Switching Interaction Models

We present the **state-space switching interaction model (SSIM)** that allows Bayesian discrete-time interaction analysis and accounts for switching interactions, as well as noisy and missing observations. We extend the basic model to include a variant in which interactions switch independently at the object (rather than group) level. SSIM is based on the assumptions that we made in Chapter 1 that allow us to represent an interaction as a structure of a dynamic Bayesian network (DBN). Therefore, SSIM can be viewed as a framework for Bayesian learning of a structure of a switching DBN from imperfect data. This problem of structure inference is hard (see Section 2.6.3). In fact, even learning a structure of a homogenous (non-switching) DBN from perfect time-series data is NP hard in general [10]. Moreover, for a fixed time-window of data, the uncertainty about the correct structure may grow with the number of time-series involved in an interaction since the number of possible structures grows super-exponentially with the number of time-series (Theorem 2.6.2) and there could possibly be many structures that explain the data well (i.e., that result in a model for which the likelihood of the data is high). The problem is further exacerbated by allowing an interaction to switch over time and by having noisy and missing data.

We will incorporate uncertainty using a Bayesian approach, in which we compute the posterior distribution over interactions, switching pattern and latent time-series. This allows us to characterize uncertainty and formulate various analyses as probabilistic events, such as "What is the probability of an edge $A \rightarrow B$ in the interaction structure at time point t?", "What is the probability of an edge $A \rightarrow C$, assuming the presence of edges $A \rightarrow B$ and $B \rightarrow C$?", "What is the probability that a change of behavior (i.e., switching) occurs within time window (t_1, t_2)?", and so on. Since inference in SSIM is in general intractable, we employ a Gibbs sampling approach (Section 2.4.1). However, in each step of the Gibbs sampler, which includes drawing samples of structures, inference will be performed exactly. To deal with the complexity of inference over structures, we employ a modular prior assumption and impose additional constraints on the structure (such as the bounded-indegree constraint), described in Section 2.6.2, that reduce the complexity to polynomial. These choices result in a tractable general inference proce-

dure for the SSIM model. However, the efficiency of particular steps of this procedure also depends on specific choices of probabilistic models that describe the evolution of time-series and the observation process. In particular, we introduce a linear Gaussian SSIM model (LG-SSIM) in which both temporal dependence and observation models are linear Gaussian. This model allows for efficient exact inference of latent time-series conditioned on other variables, which is another critical step in the sampling procedure. Finally, we use conjugate priors on parameters of the model, which further simplifies inference.

Related work is summarized in Section 3.1. In Section 3.2, we introduce the SSIM framework for Bayesian inference over switching time-series interaction structure under uncertainty, which extends the work of Siracusa and Fisher [49, 50] by allowing for noisy and missing observations of time-series. In Section 3.3, we introduce a linear Gaussian SSIM model (LG-SSIM), in which both dynamics and observation models are linear Gaussian models, thus extending Gaussian state-space switching models (e.g., [21]) to include structural inference. In this Section, we also introduce a latent-AR variant of the LG-SSIM, in which an autoregressive (AR) model of an arbitrary order is allowed among the latent state variables. Both LG-SSIM and latent-AR LG-SSIM can be paralleled to analogous extensions of the model of Siracusa and Fisher [49, 50], in which direct observations of time-series are assumed. In Section 3.4, we develop a Gibbs sampling procedure for inference in SSIM, which simultaneously reasons over interaction structures and parameters, the pattern of switching between different interactions, latent states associated with time-series, and observation model parameters. The algorithm extends the Gibbs sampling inference procedure of Siracusa and Fisher [49, 50] (Algorithm 2.1 in Section 2.7) to include steps in which latent states and observation model parameters are sampled. We also develop a specialization of the inference procedure for the LG-SSIM. In particular, we develop a numerically stable algorithm for block-sampling of latent states trajectories given observations that could be noisy and missing, and for dynamic models that allow for deterministic dependencies among state variables, such as in latent-AR LG-SSIM. Finally, we provide in-depth time and memory complexity analysis of the Gibbs sampling inference algorithm for the LG-SSIM in Section 3.5.

■ 3.1 Related Work

The proposed model integrates inference over structures, dynamic switching, and latent state-space models. All have been the subject of extensive research. Change point detection was first a subject of interest in the area of quality control, but has since become an important problem in time-series analysis domains. A huge number of online and offline, Bayesian and non-Bayesian, parametric and nonparametric methods have been developed. Basseville and Nikiforov [3] and Polunchenko and Tartakovsky [45] provide an overview of these methods. Most of these methods assume segment independence. In contrast, switching dynamic systems (SDS) – also called state-space switching mod-

els (SSM) – allow coupling between segments through dynamics parameters, which is typically modeled via latent switching states. They combine state-space modeling with switching point detection. Inference in SDS models is done via approximate methods (Pavlovic et al. [42, 43]), EM algorithm (Oh et al. [41]), or sampling (Fox et al. [16, 17]). Most of related work deals with switching linear dynamic systems (SLDS) since they allow for simpler inference but are still widely applicable.

In recent years, a number of methods for learning changing structure among time-series have been suggested. For example, Xuan and Murphy [59] combine inference over undirected graphs with change-point detection. Optimization techniques have been used to estimate time-varying undirected networks (Kolar et al. [32]), as well as time-varying DBNs (Song et al. [52]). Jiang et al. [31] use EM algorithm to obtain the MAP estimate of a switching DBN. Lebre et al. [35] and Robinson and Hartemink [46] use MCMC sampling method to learn time-varying DBNs. However, the number of sampled structures may not be sufficient to adequately represent the posterior over structures. Siracusa and Fisher [50] develop a method based on prior modularity for efficient reasoning over the structure posterior. The model we propose is most closely related to the work of [50]. It differs (in fact, from most available methods) in that we do not assume direct observation and allow for missing data. The result is a more expressive and robust model at the cost of a more complex inference procedure.

■ 3.2 SSIM Framework

The switching state-space interaction model (SSIM) is an extension of the switching temporal interaction model of Siracusa and Fisher (STIM) [49, 50] that allows for noisy observation processes. In fact, the SSIM model subsumes the STIM model, which is presented in Section 2.7. Here, we assume the notation and parts of the model introduced in Section 2.7 and only describe the difference from it.

We model that the observed value Y_t^i of signal i at time t is generated from its state X_t^i via a probabilistic observation model $p(Y_t^i|X_t^i, \xi_t^i)$ parametrized by ξ_t^i. For simplicity, we assume that the observation model is independent of the state ($\xi_t^i = \xi^i$, $\forall t, i$),

$$p(Y|X, \xi) = \prod_{t=0}^{T} \prod_{i=1}^{N} p(Y_t^i|X_t^i, \xi^i), \tag{3.1}$$

where $Y = \{Y_t\}_{t=1}^T$ is the observation sequence and ξ is the collection of parameters $\{\xi^i\}_{i=1}^N$ that describe the measurement process, including the observation noise. The model does not presume that the observation noise model is completely known (only its parametric form), and parameters ξ are also inferred.

The full SSIM generative model, shown in Figure 3.1, incorporates probabilistic models described above along with priors on structures and parameters:

- Multinomials π are drawn from Dirichlet priors parametrized by α as
  $(\pi_1, \ldots, \pi_K) \sim \mathcal{D}ir(\alpha_1, \ldots, \alpha_K)$,
  $(\pi_{i,1}, \ldots, \pi_{i,K}) \sim \mathcal{D}ir(\alpha_{i,1}, \ldots, \alpha_{i,K})$ $\forall i$.

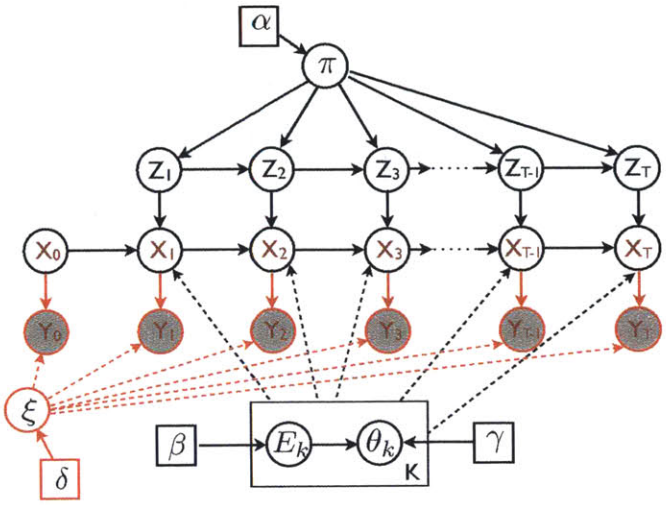


Figure 3.1: State-space switching interaction model (SSIM).

- K structures $\tilde{E}_k$ and parameters $\tilde{\theta}_k$ are drawn from the corresponding priors as $\tilde{E}_k \sim p(E;\beta)$, $\tilde{\theta}_k \sim p(\theta|\tilde{E}_k;\gamma)$, $\forall k$.
- Parameters of the observation model are drawn as $\xi^i \sim p(\xi^i;\delta)$, $\forall i$.
- Initial values X_0 and Y_0 are drawn as $X_0 \sim p(X_0|\theta_0)$ and $Y_0 \sim p(Y_0|X_0,\xi)$.
- For each $t = 1, 2, \ldots, T$ (in that order), values of Z_t, X_t and Y_t are drawn as
  $Z_t \sim \mathcal{M}ult(\pi_{Z_{t-1},1}, \ldots, \pi_{Z_{t-1},K})$ or
  $Z_t \sim \mathcal{M}ult(\pi_1, \ldots, \pi_K)$ if $t = 1$,
  $X_t \sim p(X_t|X_{t-1}, \tilde{E}_{Z_t}, \tilde{\theta}_{Z_t})$ and $Y_t \sim p(Y_t|X_t,\xi)$.

The choice of dependence and observations models is application specific and will impact the complexity of some of the inference steps. For example, commonly used linear Gaussian models (Section 3.3) allow efficient inference in state space models, which is a sub-procedure in our sampling algorithm (step 1 in Algorithm 3.1). Also, the choice of conjugate priors on parameters of dependence and observation models results in closed form expressions for sampling steps 4 and 5 in Algorithm 3.1, respectively. In this paper, we focus on linear Gaussian models and their conjugate priors, as described in Section 3.3.

Here, β are the hyperparameters of the prior on dependence structure, $p(E;\beta)$, and γ are the hyperparameters of the prior on dependence model parameters given structure, $p(\theta|E;\gamma)$. We assume that these priors are the same for all K models. Since the distribution on structure is discrete, in the most general form, β is a collection of parameters $\{\beta_E\}$ (one parameter for each structure), such that β_E is proportional to the prior probability of E:

$$p(E;\beta) = \frac{1}{B}\beta_E \propto \beta_E\,, \tag{3.2}$$

where $B = \sum_E \beta_E$ is a normalization constant. Note that the prior on parameters, $p(\theta|E;\gamma)$, may depend on the structure and γ is, in general, a collection $\{\gamma_E\}$ of sets of hyperparameters, such that $p(\theta|E;\gamma) = p(\theta;\gamma_E)$.

Learning Bayesian network structures (under reasonable assumptions) is NP hard [10]. The number of possible structures is superexponential in the number of nodes, and, in the worst case, it may be necessary to calculate the posterior of each one separately. The same holds in the case of inference of a dependence structure described above (i.e., a dependence structure of a homogenous DBN). The number of possible such structures is 2^{N^2}.

We employ two fairly general assumptions in order to reduce the complexity of inference over structures. First, we assume a modular prior on structure and parameters [7, 11, 19, 27], which decomposes as a product of priors on parent sets of individual signals and associated parameters:

$$p(E,\theta\,|\,\beta,\gamma) = \prod_{i=1}^{N} p(pa(i)\,|\,\beta)\; p(\theta^i\,|\,pa(i);\gamma)\;. \tag{3.3}$$

As a result, parent sets can be chosen independently for each signal [50], and the total number of parent sets to consider is $N2^N$, which is exponential in the number of signals. Also, β is no longer a collection of parameters per structure, but rather a collection of parameters $\{\beta_{i,pa(i)}\}$ (one parameter for each possible parent set of each signal), such that

$$p(pa(i);\beta) = \frac{1}{B_i}\beta_{i,pa(i)} \propto \beta_{i,pa(i)}\,, \tag{3.4}$$

where $B_i = \sum_s \beta_{i,s}$ are normalization constants. Modularity is also reflected in the posterior:

$$p(E,\theta\,|X;\beta,\gamma) = \prod_{i=1}^{N} p(pa(i)\,|\,X;\beta)\; p(\theta^i\,|X,pa(i);\gamma)\;. \tag{3.5}$$

If, in addition, the number of parents of each signal is bounded by some constant M (a structure with bounded in-degree [11, 19, 27]), the number of parent sets to evaluate is further reduced to $O(N^{M+1})$, which is polynomial in N.

■ 3.3 Linear Gaussian SSIM (LG-SSIM)

So far, we have described the general SSIM. Particular choices of dependence and observation models and priors may lead to specific classes of models with special properties.

Linear Gaussian state-space switching interaction models (LG-SSIM) are an instance of SSIM in which the dependence and observation models of each signal i at each time point t are linear and Gaussian:

$$\begin{aligned} X_t^i &= \tilde{A}^i_{Z_t} X_{t-1}^{\tilde{pa}(i,Z_t)} + w_t^i\,, \qquad w_t^i \sim \mathcal{N}(0,\tilde{Q}^i_{Z_t}) \\ Y_t^i &= C^i X_t^i + v_t^i\,, \qquad\qquad v_t^i \sim \mathcal{N}(0,R^i)\,. \end{aligned} \tag{3.6}$$

$\tilde{A}^i_k$ and $\tilde{Q}^i_k$ are the dependence matrix and the noise covariance matrix of signal i in the k^{th} dependence model (i.e., $\tilde{\theta}^i_k = (\tilde{A}^i_k, \tilde{Q}^i_k)$), while C^i and R^i are the observation matrix and the noise covariance matrix of the observation model of signal i (i.e., $\xi^i = (C^i, R^i)$). In addition, we assume that the value of the joint latent state at time 0 (initial state) is drawn from a Gaussian distribution with mean μ_0 and covariance Q_0:

$$X_0 \sim \mathcal{N}(\mu_0, Q_0)\,. \tag{3.7}$$

We utilize the well known matrix normal inverse Wishart distribution (Section 2.2.6) as a conjugate prior on the parameters (A, Q) of a linear Gaussian model:

$$p(A, Q; M, \Omega, \kappa, \Psi) = \mathcal{MN}(A; M, \Omega, Q)\,\mathcal{IW}(Q; \kappa, \Psi)\,. \tag{3.8}$$

Here, κ and Ψ are the degree of freedom and the inverse scale matrix parameters of the inverse Wishart distribution, while M, Ω and Q are the mean, the row covariance and the column covariance parameters of the matrix normal distribution. Note that the two distributions are coupled. The matrix normal distribution of the parameter A depends on the parameter Q that is sampled from the inverse Wishart distribution.

Recall that, due to the prior modularity assumption, for each signal i there can be a different prior on dependence model parameters for each possible parent set, and, in general, for each of the K (switching) dependence models:

$$\begin{aligned} &p(\tilde{A}^i_k, \tilde{Q}^i_k \,|\, \tilde{pa}(i,k); M^{i,\tilde{pa}(i,k)}_k, \Omega^{i,\tilde{pa}(i,k)}_k, \kappa^{i,\tilde{pa}(i,k)}_k, \Psi^{i,\tilde{pa}(i,k)}_k) \\ &= \mathcal{MN}(\tilde{A}^i_k; M^{i,\tilde{pa}(i,k)}_k, \Omega^{i,\tilde{pa}(i,k)}_k, \tilde{Q}^i_k)\,\mathcal{IW}(\tilde{Q}^i_k; \kappa^{i,\tilde{pa}(i,k)}_k, \Psi^{i,\tilde{pa}(i,k)}_k)\,. \end{aligned} \tag{3.9}$$

Throughout this thesis, we will assume that, for each signal i, C^i is known for each particular application and can be treated as a constant. For example, if a signal represents a 2D object whose noisy position is observed over time, while the state space associated wth that signal is defined as its 3^{rd} order kinematic state (position, velocity and acceleration), then

$$X^i_t = \begin{bmatrix} p^i_{xt} \\ p^i_{yt} \\ v^i_{xt} \\ v^i_{yt} \\ a^i_{xt} \\ a^i_{yt} \end{bmatrix} \qquad Y^i_t = \begin{bmatrix} \mathring{p}^i_{xt} \\ \mathring{p}^i_{yt} \end{bmatrix} \qquad C^i = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{3.10}$$

where p, v and a stand for *position*, *velocity* and *acceleration*, respectively, x and y indices refer to x and y coordinates, while $\mathring{p}^i_{xt}$ and $\mathring{p}^i_{yt}$ are noisy observations of p^i_{xt} and p^i_{yt}, respectively. We this assumption, we only need a prior on the observation covariance matrix, R^i,

$$p(R^i; \kappa^i_R, \Psi^i_R) = \mathcal{IW}(R^i; \kappa^i_R, \Psi^i_R)\,. \tag{3.11}$$

Furthermore, if the signals are uniform in representation and the observation process is the same across signals, then the model in which signals share their observation model parameters is suitable:

$$Y_t^i = C^0 X_t^i + v_t^i, \qquad v_t^i \sim \mathcal{N}(0, R^0). \tag{3.12}$$

In that case, the prior on the observation covariance matrix is

$$p(R^0; \kappa_R^0, \Psi_R^0) = \mathcal{IW}(R^0; \kappa_R^0, \Psi_R^0). \tag{3.13}$$

For example, if signals represent people whose positions are estimated (e.g., by a tracker) or annotated (e.g., by a same person), then the same observation model can be assumed for all signals.

The assumption that C^i matrices are known is made for two reasons. First, prior knowledge of any model parameters reduces the complexity of the space of solutions and therefore removes part of uncertainty in the inference result. Second, fixing the definition of latent state variables helps interpret the result of interaction inference. Note that, regardless of whether the meaning of latent state variables is predefined or not, it is important that they are related to the observations and that the complexity of the latent state is controlled such that it does not allow for arbitrary (overfitting) explanations. Besides the connection to the observations, (deterministic and probabilistic) constraints among latent state variables also reduce the complexity of the latent space. For example, if objects are represented by their kinematic state, equations of motion must be encoded into the model in order to maintain that representation, which, in turn, controls the complexity of that space.

We will also assume that the parameters of the initial state model, μ_0 and Q_0, are given, hence there will be no prior distribution on these two parameters.[1] One reason for doing this is that we do not aim at learning the initial state distribution, but rather at learning the (time-varying) dependence model. Another reason is that in most (if not all) experiments in this thesis, we will deal with a single observation sequence. Therefore, there will only be a single data sample for the initial state, which is not sufficient to learn the initial distribution. On the other hand, in cases when there are multiple observation sequences, learning the initial distribution would be plausible.

Finally, it is sometimes convenient to look at the dependence and observation models at the level of all signal jointly. For that purpose, we assume that the joint latent state, X_t, is a vector that is obtained by concatenating the latent states of signals:

$$X_t = \begin{bmatrix} X_t^1 \\ X_t^2 \\ \vdots \\ X_t^N \end{bmatrix}. \tag{3.14}$$

[1] Technically, this can be thought of as putting a degenerate prior distribution on μ_0 and Q_0 that has all probability mass on given values.

The **joint dependence model** can now be written as

$$X_t = \tilde{A}_{Z_t} X_{t-1} + w_t\,, \qquad w_t \sim \mathcal{N}(0, \tilde{Q}_{Z_t})\,, \tag{3.15}$$

where

$$\tilde{A}_{Z_t} = \begin{bmatrix} \tilde{A}'^1_{Z_t} \\ \tilde{A}'^2_{Z_t} \\ \vdots \\ \tilde{A}'^N_{Z_t} \end{bmatrix}\,, \qquad \tilde{Q}_{Z_t} = \begin{bmatrix} \tilde{Q}^1_{Z_t} & 0 & \cdots & 0 \\ 0 & \tilde{Q}^2_{Z_t} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \tilde{Q}^N_{Z_t} \end{bmatrix}\,, \tag{3.16}$$

and $\tilde{A}'^i_{Z_t}$ is such that its columns that correspond to $\tilde{pa}(i, Z_t)$ are equal to the columns of $\tilde{A}^i_{Z_t}$, while its other columns are equal to 0 (in other words, $\tilde{A}'^i_{Z_t}$ is an "expanded" version of $\tilde{A}^i_{Z_t}$ that is multiplied by X_{t-1} to predict X^i_t, i.e., $\tilde{A}'^i_{Z_t} X_{t-1} = \tilde{A}^i_{Z_t} X^{\tilde{pa}(i,Z_t)}_{t-1}$). Similarly, the **joint observation model** can be written as

$$Y_t = C\, X_t + v_t\,, \qquad v_t \sim \mathcal{N}(0, R)\,, \tag{3.17}$$

where

$$C = \begin{bmatrix} C^1 & 0 & \cdots & 0 \\ 0 & C^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & C^N \end{bmatrix}\,, \qquad R = \begin{bmatrix} R^1 & 0 & \cdots & 0 \\ 0 & R^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & R^N \end{bmatrix}\,. \tag{3.18}$$

■ 3.3.1 Latent autoregressive LG-SSIM

The LG-SSIM model above implies a first order Markov process in the latent space. However, it extends to a higher, r^{th} order process by defining a new state at time t as

$$X'_t = \begin{bmatrix} X_t \\ X_{t-1} \\ \vdots \\ X_{t-r+1} \end{bmatrix}\,, \tag{3.19}$$

i.e., by incorporating a history of length r as a basis for predicting a state at time $t+1$. Thus, an r^{th} order autoregressive model among states X_t,

$$X_t = A_{t1}\, X_{t-1} + \ldots + A_{tr}\, X_{t-r} + w_t\,, \quad w_t \sim \mathcal{N}(0, Q_t)\,, \tag{3.20}$$

transforms into a first-order AR model among states X'_t:

$$\overbrace{\begin{bmatrix} X_t \\ X_{t-1} \\ X_{t-2} \\ \vdots \\ X_{t-r+2} \\ X_{t-r+1} \end{bmatrix}}^{X'_t} = \overbrace{\begin{bmatrix} A_{t1} & A_{t2} & A_{t3} & \cdots & A_{tr-1} & A_{tr} \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{bmatrix}}^{A'_t} \overbrace{\begin{bmatrix} X_{t-1} \\ X_{t-2} \\ X_{t-3} \\ \vdots \\ X_{t-r+1} \\ X_{t-r} \end{bmatrix}}^{X'_{t-1}} + \overbrace{\begin{bmatrix} w_t \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}}^{w'_t}, \tag{3.21}$$

where

$$w'_t \sim \mathcal{N}(0, Q'_t), \quad Q'_t = \begin{bmatrix} Q_t & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}. \tag{3.22}$$

We will refer to this model as a **latent autoregressive LG-SSIM (Latent-AR LG-SSIM)** of AR order r, since the autoregressive modeling is done in the latent space. The matrix A'_t has a specific form: the first row consists of (matrix-valued) coefficients of the AR model of X_t, subdiagonal entries equal 1 and the rest of the matrix is filled with zeroes. Subdiagonal "ones" serve to simply copy the history of the expanded state, $X_{t-1}, \ldots, X_{t-r+1}$, from the expanded state at the previous time point. Therefore, the relationship between X'_t and X'_{t-1} is partially deterministic, which is reflected in the structure of the matrix Q'_t – only the first block (the one corresponding to the noisy relationship of X_t to the past) is non-zero. Thus, Q'_t is a singular matrix, and the Gaussian distribution of w'_t above is degenerate, as long as the order of the AR model is higher than 1. This is important to have in mind when developing inference algorithms for the Latent-AR LG-SSIM model, as we will discuss in Section 3.4.2.

Note that the latent-AR extension of the SSIM model, as given in Equation 3.21, is pertinent to LG-SSIM due to the linearity and Gaussianity assumptions. However, the state expansion of Equation 3.19 results in an r^{th} order latent Markov process in any SSIM model – just, the dependence model of X'_t may have a different form.

Finally, the observation model in Latent-AR LG-SSIM allows for the observation of signal i at time t, Y_t^i, to be a linear function (up to Gaussian noise) of the expanded state, X'^i_t:

$$Y_t^i = C'^i X'^i_t + v_t^i, \quad v_t^i \sim \mathcal{N}(0, R^i). \tag{3.23}$$

In other words, Y_t^i can depend on the original state at time t, X_t^i, as well as its value at the previous $r-1$ time points. Still, in all cases considered in this thesis, Y_t^i will depend only on the instant value of the original state X_t^i, as in Equation 3.6,[2] which

[2]Recall that the expanded state is introduced artificially in order to model higher order dependencies in the latent space.

can be written as:

$$Y_t^i = \overbrace{\begin{bmatrix} C^i & 0 & \dots & 0 \end{bmatrix}}^{C} \overbrace{\begin{bmatrix} X_t^i \\ X_{t-1}^i \\ X_{t-2}^i \\ \vdots \\ X_{t-r+2}^i \\ X_{t-r+1}^i \end{bmatrix}}^{X_t'^i} + v_t^i, \quad v_t^i \sim \mathcal{N}(0, R^i). \tag{3.24}$$

The joint observation model can be written as

$$Y_t = C' X_t' + v_t, \quad v_t \sim \mathcal{N}(0, R), \tag{3.25}$$

and, in the case of a dependence on the original state only,

$$Y_t = \overbrace{\begin{bmatrix} C & 0 & \dots & 0 \end{bmatrix}}^{C'} \overbrace{\begin{bmatrix} X_t \\ X_{t-1} \\ X_{t-2} \\ \vdots \\ X_{t-r+2} \\ X_{t-r+1} \end{bmatrix}}^{X_t'} + v_t, \quad v_t \sim \mathcal{N}(0, R). \tag{3.26}$$

■ 3.4 Gibbs Sampling Inference

Exact inference for the SSIM is generally intractable. Consequently, we develop a Gibbs sampling procedure as described in Algorithm 3.1, which extends the inference algorithm of Siracusa and Fisher [50], described as Algorithm 2.1 in Section 2.7, with the steps in which latent states and parameters of the observation model are sampled (steps 1 and 5, respectively).

Algorithm 3.1 SSIM Gibbs sampler

1. $X \sim p(X|Z, Y, \tilde{E}, \tilde{\theta}, \xi)$
2. $Z \sim p(Z|X, \tilde{E}, \tilde{\theta}, \pi)$
3. $\pi \sim p(\pi|Z; \alpha)$
4. $\tilde{E}, \tilde{\theta} \sim p(\tilde{E}, \tilde{\theta}|Z, X; \beta, \gamma)$
5. $\xi \sim p(\xi|X, Y; \delta)$

Note that the steps 2, 3 and 4 are the same as in Algorithm 2.1 and their respective algorithms are described in detail in Section 2.7. The complexity of sampling parameters

ξ (step 5) depends on the particular choice of the observation model. When a conjugate prior is used, this step is similarly straightforward. This is the case in LG-SSIM, in which an inverse-Wishart prior on the observation noise covariance matrix is chosen, as it is a conjugate distribution for the Gaussian distribution with a known mean. The procedure for sampling the observation noise covariance matrix in LG-SSIM that assumes an observation model shared across all signals (i.e., $\xi = R^0$) is shown in Algorithm 3.2. We proceed with the details of step 1, which is the most complicated part of the inference procedure in the SSIM model.

Algorithm 3.2 Sampling of the observation model parameters in LG-SSIM with uniform observation model and known observation matrix: $R^0 \sim p(R^0|X, Y; \kappa_R^0, \Psi_R^0)$

Inverse Wishart prior conjugate update

$\kappa'^0_R = \kappa^0_R$

$\Psi'^0_R = \Psi^0_R$

for $t = 0, \ldots, T$

  for $i = 1, \ldots, N$

    if Y_t^i *is observed*

      $\kappa'^0_R = \kappa'^0_R + 1$

      $\Psi'^0_R = \Psi'^0_R + \left(Y_t^i - C^0 X_t^i\right)\left(Y_t^i - C^0 X_t^i\right)^T$

Sampling observation noise covariance

$R^0 \sim \mathcal{IW}(R^0; \kappa'^0_R, \Psi'^0_R)$

■ 3.4.1 Batch sampling of the state sequence (step 1)

Conceptually, sampling a state sequence X when all other variables in the model are known can be performed via the same backward message-passing forward sampling algorithm as in step 2. Similar to Z, a conditional distribution of X can be decomposed as

$$P(X|Z, Y, \tilde{E}, \tilde{\theta}, \xi) = P(X_0|Z, Y, \tilde{E}, \tilde{\theta}, \xi) \prod_{t=1}^{T} P(X_t|X_{0:t-1}, Z, Y, \tilde{E}, \tilde{\theta}, \xi), \tag{3.27}$$

suggesting the following forward sampling procedure:

$$\begin{aligned} X_0 &\sim P(X_0|Z, Y, \tilde{E}, \tilde{\theta}, \xi) \\ X_1 &\sim P(X_1|X_0, Z, Y, \tilde{E}, \tilde{\theta}, \xi) \\ &\cdots \\ X_T &\sim P(X_T|X_{0:T-1}, Z, Y, \tilde{E}, \tilde{\theta}, \xi). \end{aligned} \tag{3.28}$$

$P(X_0|Z,Y,\tilde{E},\tilde{\theta},\xi)$ can be computed in the following way:

$$
\begin{aligned}
&P(X_0|Z,Y,\tilde{E},\tilde{\theta},\xi) \\
&\propto P(X_0,Y|Z,\tilde{E},\tilde{\theta},\xi) \\
&= \int_{X_{1:T}} P(X_0,X_{1:T},Y|Z,\tilde{E},\tilde{\theta},\xi)\,dX_{1:T} \\
&= P(X_0|\theta_0)P(Y_0|X_0,\xi)\int_{X_{1:T}}\left[\prod_{t=1}^{T}P(X_t|X_{t-1},\tilde{E}_{Z_t},\tilde{\theta}_{Z_t})P(Y_t|X_t,\xi)\right]dX_{1:T} \\
&= P(X_0|\theta_0)P(Y_0|X_0,\xi)\int_{X_{1:T-1}}\left[\prod_{t=1}^{T-1}P(X_t|X_{t-1},\tilde{E}_{Z_t},\tilde{\theta}_{Z_t})P(Y_t|X_t,\xi)\right]dX_{1:T-1} \\
&\qquad\qquad \times \underbrace{\int_{X_T} P(X_T|X_{T-1},\tilde{E}_{Z_T},\tilde{\theta}_{Z_T})P(Y_T|X_T,\xi)\,dX_T}_{m^{T-1}(X_{T-1})} \\
&= P(X_0|\theta_0)P(Y_0|X_0,\xi)\int_{X_{1:T-2}}\left[\prod_{t=1}^{T-2}P(X_t|X_{t-1},\tilde{E}_{Z_t},\tilde{\theta}_{Z_t})P(Y_t|X_t,\xi)\right]dX_{1:T-2} \\
&\quad \times \underbrace{\int_{X_{T-1}} P(X_{T-1}|X_{T-2},\tilde{E}_{Z_{T-1}},\tilde{\theta}_{Z_{T-1}})P(Y_{T-1}|X_{T-1},\xi)\,m^{T-1}(X_{T-1})\,dX_{T-1}}_{m^{T-2}(X_{T-2})} \\
&\qquad\qquad \vdots \\
&= P(X_0|\theta_0)P(Y_0|X_0,\xi)\underbrace{\int_{X_1} P(X_1|X_0,\tilde{E}_{Z_1},\tilde{\theta}_{Z_1})P(Y_1|X_1,\xi)\,m^1(X_1)\,dX_1}_{m^0(X_0)} \\
&= P(X_0|\theta_0)P(Y_0|X_0,\xi)\,m^0(X_0)\,. \qquad (3.29)
\end{aligned}
$$

Messages are defined recursively as

$$
\begin{aligned}
&m^T(x) = 1, \quad \forall x \in \mathcal{R}^{ND_x} \\
&m^t(x) = \int_{X_{t+1}} P(X_{t+1}|x,\tilde{E}_{Z_{t+1}},\tilde{\theta}_{Z_{t+1}})P(Y_{t+1}|X_{t+1},\xi)\,m^{t+1}(X_{t+1})\,dX_{t+1}\,, \\
&\qquad\qquad \forall x \in \mathcal{R}^{ND_x},\ \forall t = 0,\ldots,T-1\,, \qquad (3.30)
\end{aligned}
$$

where N is the number of signals and D_x is the dimensionality of the latent state of each signal (or average dimensionality if they are not uniform). Messages can also be

defined non-recursively as

$$m^t(x) = \int_{X_{t+1:T}} P(X_{t+1}|x, \tilde{E}_{Z_{t+1}}, \tilde{\theta}_{Z_{t+1}}) \prod_{i=t+2}^{T} P(X_i|X_{i-1}, \tilde{E}_{Z_i}, \tilde{\theta}_{Z_i}) \prod_{i=t+1}^{T} P(Y_i|X_i, \xi)\ dX_{t+1:T}$$
$$\forall x \in \mathcal{R}^{ND_x},\ \forall t = 0, \ldots, T-1\,. \tag{3.31}$$

Note that the meaning of a backward message is

$$m^t(x) \propto P(Y_{t+1}, ..., Y_T \mid X_t = x, Z, \tilde{E}, \tilde{\theta}, \xi)\,. \tag{3.32}$$

Finally, $P(X_t|X_{0:t-1}, Z, Y, \tilde{E}, \tilde{\theta}, \xi)$, for each $t = 1, \ldots, T$, can be computed as:

$$\begin{aligned}
&P(X_t|X_{0:t-1}, Z, Y, \tilde{E}, \tilde{\theta}, \xi) \\
&\propto P(X_t, Y|X_{0:t-1}, Z, \tilde{E}, \tilde{\theta}, \xi) \\
&= \int_{X_{t+1:T}} P(X_t, X_{t+1:T}, Y|X_{0:t-1}, Z, \tilde{E}, \tilde{\theta}, \xi)\, dX_{t+1:T} \\
&\propto \int_{X_{t+1:T}} P(X_t, X_{t+1:T}, Y_{t:T}|X_{t-1}, Z, \tilde{E}, \tilde{\theta}, \xi)\, dX_{t+1:T} \\
&= P(X_t|X_{t-1}, \tilde{E}_{Z_t}, \tilde{\theta}_{Z_t}) P(Y_t|X_t, \xi) \int_{X_{t+1:T}} \prod_{i=t+1}^{T} P(X_i|X_{i-1}, \tilde{E}_{Z_i}, \tilde{\theta}_{Z_i}) P(Y_i|X_i, \xi)\ dX_{t+1:T} \\
&= P(X_t|X_{t-1}, \tilde{E}_{Z_t}, \tilde{\theta}_{Z_t}) P(Y_t|X_t, \xi)\, m^t(X_t)\,.
\end{aligned} \tag{3.33}$$

The derived algorithm is presented in Algorithm 3.3. In general, exact computation of messages is not possible since there is an infinite number of values to compute ($x \in \mathcal{R}^{ND_x}$), and, thus, one may need to resort to an approximate method such as particle filtering [2]. However, in some particular cases, messages have a nice functional form that can be represented with finite number of parameters,[3] resulting in exact and efficient backward message passing (and forward sampling, provided that $P(X_t|X_{0:t-1}, Z, Y, \tilde{E}, \tilde{\theta}, \xi)$ has a functional form that is easy to sample from).

■ 3.4.2 Batch sampling of the state sequence in LG-SSIM model

In LG-SSIM, as we will see, each message represents a Gaussian distribution:[4]

$$m^t(x) = \mathcal{N}(x; \mu_t^m, \Sigma_t^m)\,. \tag{3.34}$$

Therefore, computing a message reduces to computing its mean and covariance.

[3] sufficient statistics

[4] Messages, as computed by Equation 3.30, are only proportional to a Gaussian distribution, but they can be normalized after each step.

Algorithm 3.3 Batch sampling of the state sequence: $X \sim p(X|Z,Y,\tilde{E},\tilde{\theta},\xi)$

Backward message passing

$m^T(x) = 1, \ \forall x$

for $t = T-1,\ldots,0$

$$m^t(x) = \int_{X_{t+1}} P(X_{t+1}|x,\tilde{E}_{Z_{t+1}},\tilde{\theta}_{Z_{t+1}})P(Y_{t+1}|X_{t+1},\xi)\, m^{t+1}(X_{t+1})\, dX_{t+1}\,, \quad \forall x$$

Forward sampling

$P(X_0|Z,Y,\tilde{E},\tilde{\theta},\xi) \propto P(X_0|\theta_0)P(Y_0|X_0,\xi)\, m^0(X_0)$

$X_0 \sim P(X_0|Z,Y,\tilde{E},\tilde{\theta},\xi)$

for $t = 1,\ldots,T$

$P(X_t|X_{0:t-1},Z,Y,\tilde{E},\tilde{\theta},\xi) \propto P(X_t|X_{t-1},\tilde{E}_{Z_t},\tilde{\theta}_{Z_t})P(Y_t|X_t,\xi)\, m^t(X_t)$

$X_t \sim P(X_t|X_{0:t-1},Z,Y,\tilde{E},\tilde{\theta},\xi)$

Recall that in LG-SSIM the joint dependence model is

$$X_t = \tilde{A}_{Z_t} X_{t-1} + w_t\,, \quad w_t \sim \mathcal{N}(0,\tilde{Q}_{Z_t})\,, \quad t = 1,\ldots,T, \tag{3.35}$$

where $\tilde{A}_{Z_t}$ and $\tilde{Q}_{Z_t}$ have the form given in Equation 3.16, the joint observation model is

$$Y_t = C\, X_t + v_t\,, \qquad v_t \sim \mathcal{N}(0,R)\,, \quad t = 0,\ldots,T, \tag{3.36}$$

where C and R have the form given in Equation 3.18, and the initial state model is

$$X_0 \sim \mathcal{N}(\mu_0,Q_0)\,. \tag{3.37}$$

Our algorithm allows defining the distribution of X_0 to be improper uniform distribution

$$P(X_0) \propto const, \quad \forall X_0 \in \mathcal{R}^n, \tag{3.38}$$

which is obtained by setting the inverse covariance, ${Q_0}^{-1}$, to 0.

Since matrices $\tilde{A}_k$, $\tilde{Q}_k$, C, and R are assumed known in this inference step, the assumption that there is only a small set of different such matrices that switch over time in not critical. Therefore, we will consider a more general model here, in which these matrices can possibly be different at each time point:

$$\begin{aligned} X_0 &\sim \mathcal{N}(\mu_0,Q_0)\,, \\ X_t &= A_t\, X_{t-1} + w_t\,, \quad w_t \sim \mathcal{N}(0,Q_t)\,, \quad t = 1,\ldots,T, \\ Y_t &= C_t\, X_t + v_t\,, \qquad v_t \sim \mathcal{N}(0,R_t)\,, \quad t = 0,\ldots,T, \end{aligned} \tag{3.39}$$

and keep in mind that $A_t = \tilde{A}_{Z_t}$, $Q_t = \tilde{Q}_{Z_t}$, $C_t = C$ and $R_t = R$ in LG-SSIM. Note that Equation 3.39 represents a standard linear Gaussian state-space model.

In this model, messages have the form of a Gaussian distribution (Equation 3.34) whose mean and covariance parameters can be computed recursively as (see Appendix A for derivation):

$$\begin{aligned} (\Sigma_T^m)^{-1} &= 0 \\ (\Sigma_T^m)^{-1}\mu_T^m &= 0\,, \end{aligned} \tag{3.40}$$

which is equivalent to $m^T(x) \propto 1$, and, for $t = 0, \ldots, T-1$,

$$\begin{aligned} (\Sigma_t^m)^{-1} &= A_{t+1}^T(Q_{t+1}^{-1} - Q_{t+1}^{-1}\Sigma_t^* Q_{t+1}^{-1})A_{t+1} \\ (\Sigma_t^m)^{-1}\mu_t^m &= A_{t+1}^T Q_{t+1}^{-1}\Sigma_t^*\Sigma_t^{\circ -1}\mu_t^\circ\,, \end{aligned} \tag{3.41}$$

where

$$\begin{aligned} \Sigma_t^{\circ -1} &= C_{t+1}^T R_{t+1}^{-1} C_{t+1} + {\Sigma_{t+1}^m}^{-1} \\ \Sigma_t^{\circ -1}\mu_t^\circ &= C_{t+1}^T R_{t+1}^{-1} Y_{t+1} + {\Sigma_{t+1}^m}^{-1}\mu_{t+1}^m \\ \Sigma_t^{*-1} &= Q_{t+1}^{-1} + \Sigma_t^{\circ -1}\,. \end{aligned} \tag{3.42}$$

Note that these are standard information filter recursive equations (e.g., as in Fox et al. [16]). In particular, $\rho = \Sigma^{-1}\mu$ and $\Lambda = \Sigma^{-1}$ can be used equivalently to parametrize a Gaussian distribution $\mathcal{N}(\mu, \Sigma)$, where Λ, the inverse of the covariance matrix, is called a precision matrix. Therefore, we could have written the above recursive equations in terms of $\Lambda_t^m = (\Sigma_t^m)^{-1}$ and $\rho_t^m = (\Sigma_t^m)^{-1}\mu_t^m$, which are indeed the values being computed. However, we choose to explicitely use terms $(\Sigma_t^m)^{-1}$ and $(\Sigma_t^m)^{-1}\mu_t^m$ in order to make their meaning clearer, even though μ_t^m and Σ_t^m are never computed explicitely. One advantage of the information filter form of update equations is that it is easy to represent complete uncertainty (missing information) about the variable of interest or some parts of it (assuming that it is a vector). For example, the initial message, $m^T(x)$, represents an improper Gaussian distribution – with infinite variance on all components of x, which is easily encoded by setting the inverse covariance of the message to 0.

Finally, $P(X_0|Z, Y, \tilde{E}, \tilde{\theta}, \xi)$ and $P(X_t|X_{0:t-1}, Z, Y, \tilde{E}, \tilde{\theta}, \xi)$ are also Gaussian distributions in LG-SSIM, allowing for computationally efficient forward sampling equations:

$$\begin{aligned} P(X_0|Z, Y, \tilde{E}, \tilde{\theta}, \xi) &= \mathcal{N}(X_0; \mu_0', \Sigma_0') \\ {\Sigma'}_0^{-1} &= C_0^T R_0^{-1} C_0 + (\Sigma_0^m)^{-1} \\ \mu_0' &= \Sigma_0' \left[C_0^T R_0^{-1} Y_0 + (\Sigma_0^m)^{-1}\mu_0^m\right], \end{aligned} \tag{3.43}$$

and, for $t = 1, \ldots, T$,

$$\begin{aligned} P(X_t|X_{0:t-1}, Z, Y, \tilde{E}, \tilde{\theta}, \xi) &= \mathcal{N}(X_t; \mu_t', \Sigma_t') \\ {\Sigma'}_t^{-1} &= Q_t^{-1} + C_t^T R_t^{-1} C_t + (\Sigma_t^m)^{-1} \\ \mu_t' &= \Sigma_t' \left[Q_t^{-1} A_t X_{t-1} + C_t^T R_t^{-1} Y_t + (\Sigma_t^m)^{-1}\mu_t^m\right]. \end{aligned} \tag{3.44}$$

A summary of the algorithm for batch sampling of the state sequence X in a linear Gaussian state-space model (Eq. 3.39) when parameters μ_0, Q_0, $A_1, \ldots, A_T$, $Q_1, \ldots, Q_T$, $C_0, \ldots, C_T$, $R_0, \ldots, R_T$ are given is shown in Algorithm 3.4.

Note that missing observations require only a slight modification of the algorithm. Namely, for each $t = 0, \ldots, T$, rows of matrix C_t corresponding to missing observations at time t should be set to zero.

Algorithm 3.4 Batch sampling of the state sequence in a linear Gaussian state-space model: $X \sim p(X \mid \mu_0, Q_0, A_{1:T}, Q_{1:T}, C_{0:T}, R_{0:T})$

Backward message passing

$(\Sigma_T^m)^{-1} = 0, \quad (\Sigma_T^m)^{-1}\mu_T^m = 0 \quad (i.e.,\ m^T(x) \propto 1)$

for $t = T-1, \ldots, 0$

$\Sigma_t^{\circ -1} = C_{t+1}^T R_{t+1}^{-1} C_{t+1} + {\Sigma_{t+1}^m}^{-1}$

$\Sigma_t^{\circ -1}\mu_t^\circ = C_{t+1}^T R_{t+1}^{-1} Y_{t+1} + {\Sigma_{t+1}^m}^{-1}\mu_{t+1}^m$

$\Sigma_t^{*-1} = Q_{t+1}^{-1} + \Sigma_t^{\circ -1}$

$(\Sigma_t^m)^{-1} = A_{t+1}^T (Q_{t+1}^{-1} - Q_{t+1}^{-1}\Sigma_t^* Q_{t+1}^{-1}) A_{t+1}$

$(\Sigma_t^m)^{-1}\mu_t^m = A_{t+1}^T Q_{t+1}^{-1}\Sigma_t^* \Sigma_t^{\circ -1}\mu_t^\circ$

$m^t(x) = \left((\Sigma_t^m)^{-1}\mu_t^m, (\Sigma_t^m)^{-1}\right) \equiv \mathcal{N}(x; \mu_t^m, \Sigma_t^m)$

Forward sampling

${\Sigma'_0}^{-1} = C_0^T R_0^{-1} C_0 + (\Sigma_0^m)^{-1}$

$\mu'_0 = \Sigma'_0 \left[C_0^T R_0^{-1} Y_0 + (\Sigma_0^m)^{-1}\mu_0^m\right]$

$X_0 \sim \mathcal{N}(X_0; \mu'_0, \Sigma'_0)$

for $t = 1, \ldots, T$

${\Sigma'_t}^{-1} = Q_t^{-1} + C_t^T R_t^{-1} C_t + (\Sigma_t^m)^{-1}$

$\mu'_t = \Sigma'_t \left[Q_t^{-1} A_t X_{t-1} + C_t^T R_t^{-1} Y_t + (\Sigma_t^m)^{-1}\mu_t^m\right]$

$X_t \sim \mathcal{N}(X_t; \mu'_t, \Sigma'_t)$

Algorithm with improved numerical stability

For long sequences of missing data, Σ_t^m approaches Q_{t+1} and intermediate values $Q_{t+1}^{-1} - Q_{t+1}^{-1}\Sigma_t^* Q_{t+1}^{-1}$ are close to singular. In addition, we want to allow Q_t matrices to be singular, which is for example the case in the latent-AR LGSSIM. Algorithm 3.4 however requires inversion of these matrices and is therefore unusable in this case.[5]

[5]Pseudoinverses could possibly be used, and that would require verifying the correcness of calculations.

Via the matrix equality $(A+B)^{-1} = A^{-1} - (I + A^{-1}B)^{-1}A^{-1}BA^{-1}$, we derive alternative recursive equations that yields a numerically stable algorithm and allows for singular covariance matrices, which we exploit to impose deterministic constraints between variables across time:

$$\begin{aligned}(\Sigma_t^m)^{-1} &= A_{t+1}^T \Sigma_t^\Delta \, \Sigma_t^{\circ -1} A_{t+1} \\ (\Sigma_t^m)^{-1}\mu_t^m &= A_{t+1}^T \Sigma_t^\Delta \, \Sigma_t^{\circ -1} \mu_t^\circ ,\end{aligned} \tag{3.45}$$

where

$$\begin{aligned}\Sigma_t^{\circ -1} &= C_{t+1}^T R_{t+1}^{-1} C_{t+1} + {\Sigma_{t+1}^m}^{-1} \\ \Sigma_t^{\circ -1}\mu_t^\circ &= C_{t+1}^T R_{t+1}^{-1} Y_{t+1} + {\Sigma_{t+1}^m}^{-1}\mu_{t+1}^m \\ \Sigma_t^\Delta &= (I + \Sigma_t^{\circ -1} Q_{t+1})^{-1} .\end{aligned} \tag{3.46}$$

Similarly, we derive equations for the mean μ_t' and the covariance matrix Σ_t' in the forward sampling procedure that do not require inversion of dependence covariance matrices:

$$\begin{aligned}&\mu_t' = G_t \left(G_t^{-1}\mu_t'\right) , \quad \Sigma_t' = G_t \, Q_t , \qquad \text{where} \\ &G_t^{-1} = I + Q_t \, C_t^T R_t^{-1} C_t + Q_t \, (\Sigma_t^m)^{-1} \\ &G_t^{-1}\mu_t' = A_t X_{t-1} + Q_t \, C_t^T R_t^{-1} Y_t + Q_t \, (\Sigma_t^m)^{-1}\mu_t^m .\end{aligned} \tag{3.47}$$

The above procedure is summarized in in Algorithm 3.5.

■ 3.5 Algorithmic Complexity

We analyze the time and memory complexity of each step of the Gibbs sampling algorithm (Algorithm 3.1) for inference in the LG-SSIM model in terms of various problem parameters.

Table 3.1 contains a description of problem parameters that govern the complexity of inference in the LG-SSIM model. We assume here for simplicity that all signals have the same observed and latent dimensionality. Also, we assume the latent-AR extension of LG-SSIM and include the order of the latent AR model, R, as a parameter of interest, while the latent dimensionality of a signal refers to its dimensionality prior to state expansion (i.e., the one inherent to a single time point). If the basic LG-SSIM model is considered instead, R should be ignored (or, equivalently, treated as $R = 1$). In addition, a modular bounded-indegree prior on interactions is assumed, where M is the maximum number of parents per signal allowed. Note that the number of signals, N, their observed dimensionality, D_y, and the sequence length, T, are determined purely by the data that is an input to the algorithm. On the other hand, the latent dimensionality of signals, D_x, the number of switching structures, K, the maximum number of parents per signal, M, and the latent-AR order, R, can be set arbitrarily (to some extent) in

Algorithm 3.5 Numerically stable batch sampling of the state sequence in a linear Gaussian state-space model: $X \sim p(X \mid \mu_0, Q_0, A_{1:T}, Q_{1:T}, C_{0:T}, R_{0:T})$

Backward message passing

$(\Sigma_T^m)^{-1} = 0, \quad (\Sigma_T^m)^{-1}\mu_T^m = 0 \quad (i.e.,\ m^T(x) \propto 1)$

for $t = T-1, \ldots, 0$

$\Sigma_t^{\circ -1} = C_{t+1}^T R_{t+1}^{-1} C_{t+1} + {\Sigma_{t+1}^m}^{-1}$

$\Sigma_t^{\circ -1}\mu_t^\circ = C_{t+1}^T R_{t+1}^{-1} Y_{t+1} + {\Sigma_{t+1}^m}^{-1}\mu_{t+1}^m$

$\Sigma_t^\Delta = (I + \Sigma_t^{\circ -1} Q_{t+1})^{-1}$

$(\Sigma_t^m)^{-1} = A_{t+1}^T \Sigma_t^\Delta \Sigma_t^{\circ -1} A_{t+1}$

$(\Sigma_t^m)^{-1}\mu_t^m = A_{t+1}^T \Sigma_t^\Delta \Sigma_t^{\circ -1}\mu_t^\circ$

$m^t(x) = \left((\Sigma_t^m)^{-1}\mu_t^m, (\Sigma_t^m)^{-1}\right) \equiv \mathcal{N}(x; \mu_t^m, \Sigma_t^m)$

Forward sampling

${\Sigma'_0}^{-1} = C_0^T R_0^{-1} C_0 + (\Sigma_0^m)^{-1}$

$\mu'_0 = \Sigma'_0 \left[C_0^T R_0^{-1} Y_0 + (\Sigma_0^m)^{-1}\mu_0^m\right]$

$X_0 \sim \mathcal{N}(X_0; \mu'_0, \Sigma'_0)$

for $t = 1, \ldots, T$

$G_t^{-1} = I + Q_t C_t^T R_t^{-1} C_t + Q_t (\Sigma_t^m)^{-1}$

$G_t^{-1}\mu'_t = A_t X_{t-1} + Q_t C_t^T R_t^{-1} Y_t + Q_t (\Sigma_t^m)^{-1}\mu_t^m$

$\mu'_t = G_t \left(G_t^{-1}\mu'_t\right)$

$\Sigma'_t = G_t Q_t$

$X_t \sim \mathcal{N}(X_t; \mu'_t, \Sigma'_t)$

Param.	Meaning	Determined by
N	number of signals/objects	data
D_y	observed dim. of each signal	data
T	sequence length	data
D_x	latent dim. of each signal	data/setup
K	number of switching structures	data/setup
M	maximum number of parents	data/setup
R	order of AR model	data/setup

Table 3.1: Description of problem parameters.

the inference setup.[6] However, they should be set to best capture properties of the problem of interest and the particular data used for inference, and may therefore be influenced by the data. For example, the number of switching structures may be set to a number that exceeds our prior expectation for the possible number of different behaviors (dynamics), the maximum number of parents may be set to exceed our prior expectation on how many signals can simultaneously influence a single signal (unless it must be set lower for computational purposes), and the order of the latent AR order should be set to encompass a large enough window of history, such that important dependencies can be captured (again, as long as computational resources allow that).

The asymptotic time and memory complexities of each step of the Gibbs sampling algorithm for inference in LG-SSIM in terms of the above parameters are summarized in Tables 3.2 and 3.4, respectively, while more detailed analyses for the complexities of each step are given in the subsections below. We make a few additional assumptions here. First, we assume that $K \ll T$, such that $K^2 \leq T$ is satisfied. This is showed in the time complexity analysis of step 3. We also assume that M is not higher than a fraction of N, where the fraction constant is smaller than $1/2$, i.e., that $M/N \leq c < 1/2$, as well as that $K \max(MRD_x, D_y) \leq T$. These two assumptions have an implication for the time complexity of step 4. Table 3.3 summarizes expressions for the time complexity of this step under different conditions, as discussed in Section 3.5.4. The last row of the table refers to the assumption $M/N \leq c < 1/2$, while the assumption $K \max(MRD_x, D_y) \leq T$ simplifies the expression to the one showed in Table 3.2. In addition, we do not account for missing data here. Some of the computations may be reduced by a fraction of non-missing data, although not the ones that present bottleneck.

In terms of the time complexity, steps 1 and 4 are critical. Step 3 is dominated by step 2, step 5 is dominated by steps 1 and 4, while step 2 is dominated by steps 1 and

[6] Well, the latent dimensionality of a signal, D_x, must be set according to the choice of the observation and dependence models. Typically, these models would be chosen based on the problem description and would not be changed (or not changed often) during the experimenting phase.

SSIM Gibbs sampler	Alg. in LG-SSIM	Complexity $\Theta(\cdot)$
1. $X \sim p(X\|Z, Y, \tilde{E}, \tilde{\theta}, \xi)$	Gaussian-MP	$N^3 RD_x \left({D_y}^2 + (RD_x)^2\right) T$
2. $Z \sim p(Z\|X, \tilde{E}, \tilde{\theta}, \pi)$	discrete-MP	$K\left(RD_x{}^2 NM + K\right) T$
3. $\pi \sim p(\pi\|Z; \alpha)$	conjugate update	$T + K^2 \approx T$
4. $\tilde{E}, \tilde{\theta} \sim p(\tilde{E}, \tilde{\theta}\|Z, X; \beta, \gamma)$	conjugate update	$N\binom{N}{M}\left((MRD_x)^2 + {D_y}^2\right) T$
5. $\xi \sim p(\xi\|X, Y; \delta)$	conjugate update	$D_y^2 + D_y(D_y + RD_x)NT$

Table 3.2: Time complexity of LG-SSIM Gibbs sampling steps. Common bottlenecks are shown in red.

4 unless the number of switching states, K, is large and the number of signals, their dimensionality, and AR order are small. This is however not the case in the majority of practical scenarios, so we will focus on steps 1 and 4 as bottlenecks. For $M \leq 2$, step 1 is dominant. Otherwise, the complexity of step 4 is higher as a function of N due to a higher polynomial degree and is dominant for sufficiently large N.

Table 3.4 shows, in addition to the space required for each step of the Gibbs sampler, the space required for variables that are kept outside of these steps, i.e., variables that represent the data and the model, variables that represent the current state of the sampler (latent variables in particular). For each of the five sampling steps, only the additional required memory is analyzed (input variables are excluded as they are global to the algorithm). Step 3 along with storing Z, π and α are dominated by step 2 (recall that $K < T$). Step 5 does not introduce any new complexity. Storing X is dominated by step 1. Storing Y is also most likely dominated by step 1 as $D_y \leq N(RD_x)^2$ is true in most cases (and certainly in the experiments in this thesis). Step 4 is dominated by the requirement for storing prior parameters.[7] Finally, step 2 is in most scenarios dominated by step 1 and/or prior parameters, unless K is large and N, R and D_x are very small (in which case memory is most likely not a critical resource anyway). Thus, the common bottlenecks for running the Gibbs sampling inference algorithm in LG-SSIM are step 1 and storing parameters of the prior on structure and dependence models. If the number of data points, T is very large, then step 1 can pose a memory bottleneck. On the other hand, if the number of allowed parent sets is huge (it grows very quickly with M, even for relatively small N), then storing prior parameters is a bottleneck.

[7]Unless there is a compact way of storing these parameters, such as some parametric form. Here, we assume the general case in which each parameter can be set arbitrarily.

Condition	Complexity $\Theta(\cdot)$
general	$\leq N2^N \left((NRD_x)^2 + {D_y}^2\right) T + N2^N \left((NRD_x)^3 + {D_y}^3\right) K$
$M \geq N/2$	$N2^N \left((NRD_x)^2 + {D_y}^2\right) T + N2^N \left((NRD_x)^3 + {D_y}^3\right) K$
$M = const$	$N^{M+1} \left((RD_x)^2 + {D_y}^2\right) T + N^{M+1} \left((RD_x)^3 + {D_y}^3\right) K$
$M/N \leq c < 1/2$	$N\binom{N}{M} \left((MRD_x)^2 + {D_y}^2\right) T + N\binom{N}{M} \left((MRD_x)^3 + {D_y}^3\right) K$

Table 3.3: Time complexity of step 4 of LG-SSIM Gibbs sampling algorithm, $\tilde{E}, \tilde{\theta} \sim p(\tilde{E}, \tilde{\theta} | Z, X; \beta, \gamma)$, under different assumptions. Note that this assumptions are not disjoint; they simply represent different assumptions that are reasonable to make in different circumstances.

SSIM Gibbs sampler	Alg. in LG-SSIM	Complexity $\Theta(\cdot)$
observed sequence, Y		ND_yT
latent sequence, X		NRD_xT
switching sequence, Z		T
π and α		K^2
C, R and ξ		$(NRD_x + D_y)D_y$
prior parameters, β and γ		$N\binom{N}{M} \left((MRD_x)^2 + {D_y}^2\right) K$
1. $X \sim p(X \mid Z, Y, \tilde{E}, \tilde{\theta}, \xi)$	Gaussian-MP	$(NRD_x)^2T$
2. $Z \sim p(Z \mid X, \tilde{E}, \tilde{\theta}, \pi)$	discrete-MP	KT
3. $\pi \sim p(\pi \mid Z; \alpha)$	conjugate update	K^2
4. $\tilde{E}, \tilde{\theta} \sim p(\tilde{E}, \tilde{\theta} \mid Z, X; \beta, \gamma)$	conjugate update	$\binom{N}{M} + ND_y(MRD_x + D_y)K$
5. $\xi \sim p(\xi \mid X, Y; \delta)$	conjugate update	$(D_y + RD_x)NT$

Table 3.4: Memory complexity of LG-SSIM Gibbs sampling steps. The complexity of step 5 could be reduced to just $\Theta(D_y^2)$ of additional space (see Section 3.5.5 for discussion), but that is not critical. Common bottlenecks are shown in red.

■ 3.5.1 Complexity of inference in LG-SSIM: step 1

Algorithm 3.4 describes a procedure for batch sampling of the state sequence in the LG-SSIM model. Note that, in the latent-AR variant, the dimensionality of the latent state (jointly, over all signals) is NRD_x. Thus, transition matrices, A_t, and dependence model covariance matrices, Q_t, are of dimension $NRD_x \times NRD_x$. Consequently, the mean of each message, μ_t^m, has a dimension NRD_x, while the covariance matrix of each message, Σ_t^m, has dimension $NRD_x \times NRD_x$. Also, each observation matrix, C_t, has dimension $NRD_x \times ND_y$, and each observation noise covariance matrix, R_t has dimension $ND_y \times ND_y$.

Computing a message at each time point requires a constant number of matrix (or vector) multiplications, additions and inversions. For simplicity, we will assume a "naïve" algorithm for matrix multiplication, which runs in $\Theta(n_a n_b n_c)$ time for matrices of dimensions $n_a \times n_b$ and $n_b \times n_c$. For a square matrix of dimension $n \times n$, that complexity is $\Theta(n^3)$. It should be noted that there are algorithms for matrix multiplication that run in asymptotically lower time. For example, the famous Strassen's algorithm [54] runs in $\Theta(n^{2.807})$ time and, although numerically less stable, is occasionally used in practice. On the other hand, an algorithm with the currently lowest asymptotic complexity, due to Le Gall [33], runs in $\Theta(n^{2.373})$ time, but is impractical due to an extremely large constant factor involved. The same holds for matrix inversion as it can be reduced to matrix multiplication.

By looking at Algorithm 3.4, we can see that the operations involved in message computation that dominate computational time are multiplication and inversion of matrices of dimension $NRD_x \times NRD_x$ and multiplication of matrices of dimension $NRD_x \times ND_y$ with matrices of $ND_y \times ND_y$. Therefore, assuming naïve matrix multiplication and inversion algorithms, computing a message at any time point takes $\Theta\left((NRD_x)^3 + NRD_x(ND_y)^2\right) = \Theta\left(N^3 RD_x((RD_x)^2 + {D_y}^2)\right)$ time. Note that Algorithm 3.4 also involves inversion of matrices R_t, which takes $\Theta(N^3 {D_y}^3)$. If these matrices are different at different time points, one such inversion must be computed for each message, and the time complexity of computing a single message would be $\Theta\left((NRD_x)^3 + (NRD_x)^2 ND_y + (ND_y)^3)\right)$. However, while this is true for the general form of Algorithm 3.4, recall that the SSIM model assumes a single observation model applied at all time points, which requires only a single inversion of the observation noise covariance matrix in total, which can be ignored in the complexity analysis. This would be the case in many other applications as well, since it is not realistic to expect that the number of observation models is on the same order as the number of time points. Finally, we can conclude that the time complexity of computing all messages (over all time points) is $\Theta\left(N^3 RD_x((RD_x)^2 + {D_y}^2)T\right)$. Operations of the same complexity are required in the forward sampling part of the algorithm, and so this is as well the time complexity of the whole algorithm.

Note that in many cases, the dimensionality of an observation of a signal, D_y, will be smaller than the dimensionality of the latent state associated to that signal, RD_x. For example, this will be the case if a latent representation of an object consists of its

position, velocity and acceleration, but only its position is observed. In such cases, D_y can be ignored in the time complexity analysis, and we can say that Algorithm 3.4 runs in $\Theta\left((NRD_x)^3)T\right)$ time. However, there may be applications in which the opposite is true ($D_y > RD_x$). This would for example be the case if R is small and there are multiple observations/measurements of a signal at each time point.

The memory bottleneck of Algorithm 3.4 is storing inverse covariance matrices of messages, $(\Sigma_t^m)^{-1}$. These matrices are of dimension $NRD_x \times NRD_x$. Therefore, the total memory complexity is $\Theta\left((NRD_x)^2T\right)$.

Lastly, Algorithm 3.5, which is a numerically stable version of Algorithm 3.4 that we use in practice, requires the same types of computations (matrix multiplications and inversions) and stores the same messages as Algorithm 3.4, and thus has the same time and memory complexity.

■ 3.5.2 Complexity of inference in LG-SSIM: step 2

Algorithm 2.6 describes a procedure for batch sampling of the switching state sequence in the LG-SSIM model. Evaluating $P(X_t|X_{t-1}, \tilde{E}_k, \tilde{\theta}_k)$ for any t and k requires $\Theta({RD_x}^2NM)$ time. To see this, note that in the Latent-AR LG-SSIM model $P(X_t^i|X_{t-1}, \tilde{E}_k, \tilde{\theta}_k) = \mathcal{N}(X_t^i; \tilde{A}_k^i X_{t-1}^{\tilde{pa}(i,k)}, \tilde{Q}_k^i)$. Vector X_t^i is of length D_x, while the maximum length of vector $X_{t-1}^{\tilde{pa}(i,k)}$ is MRD_x, since M is the maximum number of parent signals and RD_x is the length of an expanded state of a signal. Therefore, matrix $\tilde{A}_k^i$ has a maximum dimension $D_x \times MRD_x$, and computing the product $\tilde{A}_k^i X_{t-1}^{\tilde{pa}(i,k)}$ takes at most $\Theta(MRD_x{}^2)$ time.[8] Evaluating the above Gaussian density takes $\Theta({D_x}^2)$ since the matrix $\tilde{Q}_k^i$ is of dimension $D_x \times D_x$. Overall, evaluating $P(X_t^i|X_{t-1}, \tilde{E}_k, \tilde{\theta}_k)$ takes $\Theta(MRD_x{}^2)$ time. Therefore, evaluating $P(X_t|X_{t-1}, \tilde{E}_k, \tilde{\theta}_k)$ for any t and k takes $\Theta(NMRD_x{}^2)$ time. Finally, evaluating $P(X_t|X_{t-1}, \tilde{E}_k, \tilde{\theta}_k)$ for all t and k takes $\Theta(KTNMRD_x{}^2)$ in total. Once these probabilities are computed, computing messages takes $\Theta(TK^2)$ time, since there are in total T messages, each message consists of K values (probabilities), and computing each value requires a summation over K terms. Finally, forward sampling takes $\Theta(TK)$ time, since at each time point the probability $P(Z_t|Z_{1:t-1}, X, \tilde{E}, \tilde{\theta})$ is computed for each of K possible values of Z_t and then sampling from that multinomial distribution also takes $\Theta(K)$ time. Therefore, the total time required for Algorithm 2.6 is $\Theta(KTNMRD_x{}^2 + TK^2 + TK) = \Theta(K(RD_x{}^2NM + K)T)$.

Storing probabilities $P(X_t|X_{t-1}, \tilde{E}_k, \tilde{\theta}_k)$ for all t and k requires $\Theta(TK)$ memory. The same holds for storing messages, since there are T messages and each consists of K values. Finally, storing samples of switching state variables, Z_t, takes $\Theta(T)$ time. Therefore, the overall memory complexity of Algorithm 2.6 is $\Theta(KT)$.

[8] We will evaluate the worst case scenario and assume that the number of parents per signal is $\Theta(M)$. One may attempt to argue that that is also the average case. However, the distribution of the number of parents is unknown and may not be uniform, for which reasons such a conclusion cannot always be drawn.

■ 3.5.3 Complexity of inference in LG-SSIM: step 3

Algorithm 2.5 describes a procedure for sampling parameters of multinomial distributions that govern the evolution of the switching sequence. This step is universal for all SSIM models. Updating Dirichlet prior distributions on the initial and transition multinomials, π, with a given switching sequence, Z, requires counting the number of times that each initial switching state[9] and each switching state transition $i \rightarrow j$ $(1 \leq i, j \leq K)$ appears in the switching sequence, and adding those counts to the prior hyperparameters (pseudocounts), α, to obtain values of hyperparameters of the posterior. This can be done in $\Theta(T)$ time, as each pair of values (Z_t, Z_{t+1}) needs to be counted. While the total number of hyperparameters is $K(K+1) = \Theta(K^2)$, only $\Theta(T)$ of them need to be updated.[10] Once the pseudocounts are updated, sampling of $K+1$ multinomial distributions from the corresponding Dirichlet distributions takes $\Theta(K^2)$ time on total, as each multinomial is K-variate.[11] Therefore, the total time complexity of this sampling step is $\Theta(T + K^2)$. Note that K is typically much smaller than T, and, in most cases, it is safe to assume $K^2 \leq T$ and thus ignore K^2 term in the time complexity analysis.

The memory complexity of this step is $\Theta(K^2)$, what is required for storing the initial and transition counts of switching states obtained from the data. This could even be reduced to $\Theta(1)$ if each individual count (at each time point) is immediately added to the appropriate pseudocount, but, again, that is not a critical part since the output multinomial distributions already take $\Theta(K^2)$ space.

■ 3.5.4 Complexity of inference in LG-SSIM: step 4

A procedure for sampling structures and parameters of switching dependence models in LG-SSIM with modular prior is given in Algorithm 2.4. Vector ${X'^s}_{t-1}$ is of length $|s|RD_x$, where $|s|$ is the number of signals in the parent set s. Therefore, multiplication ${X'^s}_{t-1}{{X'^s}_{t-1}}^T$ takes $\Theta\left((|s|RD_x)^2\right)$ time. Similarly multiplications $X_t^i{{X'^s}_{t-1}}^T$ and $X_t^i X_t^{iT}$ take $\Theta\left(D_y|s|RD_x\right)$ and $\Theta\left({D_y}^2\right)$, respectively, since vector X_t^i is of length D_y. T_k such multiplications are performed, where $T_k = |\{t : Z_t = k\}|$ is the number of time points in

[9] If there is only one data sequence, counting initial states is trivial – there is only one such state in the data. However, the algorithm allows for multiple sequences as well, in which case there would be multiple initial state appearances in the data.

[10] For simplicity, in our implementation, an array of counts of initial states and state transitions is computed from the data (in a separate function) and then added to the pseudocounts. Initializing these counts and adding them to the pseudocounts takes $\Theta(K^2)$ time, and therefore the counting step technically takes $\Theta(T + K^2)$ time in our implementation. However, that does not alter the overall time complexity of this step in the inference procedure, as will see that it is $\Theta(T + K^2)$ anyway.

[11] Generating a sample from a Dirichlet distribution is reduced to generating a sample from a gamma distribution (see [12], Theorem 4.1. on p. 594), which is obtained using a rejection sampling approach (Jonk's algorithm, see [12], p. 418). Computational time of a rejection sampling algorithm depends on the actual values of parameters of a distribution (Dirichlet distribution in this case), and therefore can vary depending on data properties. For simplicity, we assume that there is a constant bound per dimension for generating these samples in practical examples.

which k^{th} model is active, yielding a time complexity $\Theta\left(\left((|s|RD_x)^2 + D_y|s|RD_x + {D_y}^2\right)T_k\right)$. Since $D_y|s|RD_x \leq (|s|RD_x)^2 + {D_y}^2$, the term $D_y|s|RD_x$ can be ignored in the asymptotic time complexity analysis. The total time complexity of these multiplications is therefore $\Theta\left(\sum_{k=1}^{K}\sum_{i=1}^{N}\sum_{s\in\mathcal{PA}_k^i}\left((|s|RD_x)^2 + {D_y}^2\right)T_k\right)$. In addition, matrices $\Omega_{i,s}^k$, $M_{i,s}^k$ and $\Psi_{i,s}^k$, as well as their posterior-updated versions, are of dimension $|s|RD_x \times |s|RD_x$, $D_y \times |s|RD_x$ and $D_y \times D_y$, respectively. Computations with these matrices that are of highest complexity are ${\Omega_{i,s}^k}^{-1}$ and $|\Omega_{i,s}^k|$, which take $\Theta\left((|s|RD_x)^3\right)$ time, $M_{i,s}^k{\Omega_{i,s}^k}^{-1}$, which takes $\Theta\left(D_y(|s|RD_x)^2\right)$ time, and $|\Psi_{i,s}^k|$, which takes $\Theta\left({D_y}^3\right)$ time. Again, $\Theta\left(D_y(|s|RD_x)^2\right)$ term can be ignored since $D_y(|s|RD_x)^2 \leq (|s|RD_x)^3 + {D_y}^3$. Since there is a constant number of these computations in each loop iteration, their total time complexity is $\Theta\left(\sum_{k=1}^{K}\sum_{i=1}^{N}\sum_{s\in\mathcal{PA}_k^i}(|s|RD_x)^3 + {D_y}^3\right)$. Finally, since all other steps of Algorithm 2.4 are dominated by these ones, the total time complexity of this algorithm is $\Theta\left(\sum_{k=1}^{K}\sum_{i=1}^{N}\sum_{s\in\mathcal{PA}_k^i}\left((|s|RD_x)^2 + {D_y}^2\right)T_k + (|s|RD_x)^3 + {D_y}^3\right)$. Note that if $T_k \geq \max(NRD_x, D_y)$, i.e., if the number of time points assigned to each model is greater or equal to the dimensionality of expanded latent state and observation state, which holds in many applications, the overall time complexity of Algorithm 2.4 can be reduced to $\Theta\left(\sum_{k=1}^{K}\sum_{i=1}^{N}\sum_{s\in\mathcal{PA}_k^i}\left((|s|RD_x)^2 + {D_y}^2\right)T_k\right)$.

A simple bound can be obtained using inequalities $|s| \leq N$, $\forall s$, and $|\mathcal{PA}_k^i| \leq 2^N$, $\forall k, i$:

$$
\begin{aligned}
&\Theta\left(\sum_{k=1}^{K}\sum_{i=1}^{N}\sum_{s\in\mathcal{PA}_k^i}\left((|s|RD_x)^2 + {D_y}^2\right)T_k + (|s|RD_x)^3 + {D_y}^3\right) = \\
&\mathcal{O}\left(\sum_{k=1}^{K}\sum_{i=1}^{N}\sum_{s\in\mathcal{PA}_k^i}\left((NRD_x)^2 + {D_y}^2\right)T_k + (NRD_x)^3 + {D_y}^3\right) = \\
&\mathcal{O}\left(\left((NRD_x)^2 + {D_y}^2\right)\sum_{k=1}^{K}\sum_{i=1}^{N}|\mathcal{PA}_k^i|T_k + \left((NRD_x)^3 + {D_y}^3\right)\sum_{k=1}^{K}\sum_{i=1}^{N}|\mathcal{PA}_k^i|\right) = \\
&\mathcal{O}\left(\left((NRD_x)^2 + {D_y}^2\right)\sum_{k=1}^{K}\sum_{i=1}^{N}2^N T_k + \left((NRD_x)^3 + {D_y}^3\right)\sum_{k=1}^{K}\sum_{i=1}^{N}2^N\right) = \\
&\mathcal{O}\left(\left((NRD_x)^2 + {D_y}^2\right)2^N NT + \left((NRD_x)^3 + {D_y}^3\right)2^N NK\right) . \qquad (3.48)
\end{aligned}
$$

Again, in most practical cases, $T > \max(NRD_x, D_y)\,K$, and the above bound can be reduced to $\mathcal{O}\left(\left((NRD_x)^2 + {D_y}^2\right)2^N NT\right)$. This bound is asymptotically achieved when all parent sets are allowed (for all signals in all models). In that case, $|\mathcal{PA}_k^i| = 2^N$, $\forall i, k$, and at least half of the subsets have size at least $N/2$ (excluding subsets of size $N/2$ if N is even, there is the same number of subsets of size smaller than $N/2$ as the number of subsets of size larger than $N/2$, which follows from equality $\binom{N}{m} = \binom{N}{N-m}$).

Let $\mathcal{PA}^{*i}_k \subset \mathcal{PA}^i_k$ denote a set of subsets of N elements whose length is at least $N/2$. Then, $|\mathcal{PA}^{*i}_k| \geq 2^N/2$, and

$$\Theta\left(\sum_{k=1}^{K}\sum_{i=1}^{N}\sum_{s\in\mathcal{PA}^i_k}\left((|s|RD_x)^2 + {D_y}^2\right)T_k + \left(|s|RD_x\right)^3 + {D_y}^3\right) =$$

$$\Omega\left(\sum_{k=1}^{K}\sum_{i=1}^{N}\sum_{s\in\mathcal{PA}^{*i}_k}\left((|s|RD_x)^2 + {D_y}^2\right)T_k + \left(|s|RD_x\right)^3 + {D_y}^3\right) =$$

$$\Omega\left(\sum_{k=1}^{K}\sum_{i=1}^{N}\sum_{s\in\mathcal{PA}^{*i}_k}\left((N/2RD_x)^2 + {D_y}^2\right)T_k + \left(N/2RD_x\right)^3 + {D_y}^3\right) =$$

$$\Omega\left(\left((NRD_x)^2 + {D_y}^2\right)\sum_{k=1}^{K}\sum_{i=1}^{N}|\mathcal{PA}^{*i}_k|T_k + \left((NRD_x)^3 + {D_y}^3\right)\sum_{k=1}^{K}\sum_{i=1}^{N}|\mathcal{PA}^{*i}_k|\right) =$$

$$\Omega\left(\left((NRD_x)^2 + {D_y}^2\right)\sum_{k=1}^{K}\sum_{i=1}^{N}2^{N-1}T_k + \left((NRD_x)^3 + {D_y}^3\right)\sum_{k=1}^{K}\sum_{i=1}^{N}2^{N-1}\right) =$$

$$\Omega\left(\left((NRD_x)^2 + {D_y}^2\right)2^N NT + \left((NRD_x)^3 + {D_y}^3\right)2^N NK\right). \tag{3.49}$$

In fact, the same holds even if each set $\mathcal{PA}^i_k$ contains only a constant fraction of all possible parent sets. To show that, note that if $|\mathcal{PA}^i_k| = c2^N$ for some constant c $(0 \leq c \leq 1)$, then at least half of the parent sets in $\mathcal{PA}^i_k$ are of size at least $cN/2$.

A different simplification of the time complexity expression can be obtained by making an assumption that allowed parent sets of each node are the same in all K models, i.e., that $\mathcal{PA}^i_k \equiv \mathcal{PA}^i$. In that case,

$$\Theta\left(\sum_{k=1}^{K}\sum_{i=1}^{N}\sum_{s\in\mathcal{PA}^i_k}\left((|s|RD_x)^2 + {D_y}^2\right)T_k + \left(|s|RD_x\right)^3 + {D_y}^3\right) =$$

$$\Theta\left(\sum_{k=1}^{K}\sum_{i=1}^{N}\sum_{s\in\mathcal{PA}^i}\left((|s|RD_x)^2 + {D_y}^2\right)T_k + \left(|s|RD_x\right)^3 + {D_y}^3\right) =$$

$$\Theta\left(\sum_{i=1}^{N}\sum_{s\in\mathcal{PA}^i}\left((|s|RD_x)^2 + {D_y}^2\right)T + \left((|s|RD_x)^3 + {D_y}^3\right)K\right) =$$

$$\Theta\left(\sum_{i=1}^{N}\left[\left(\sum_{s\in\mathcal{PA}^i}|s|^2\right)(RD_x)^2 + |\mathcal{PA}^i|{D_y}^2\right]T + \left[\left(\sum_{s\in\mathcal{PA}^i}|s|^3\right)(RD_x)^3 + |\mathcal{PA}^i|{D_y}^3\right]K\right). \tag{3.50}$$

Let us now assume a bounded in-degree prior on parent sets with the maximum

number of parents equal M for all signals and dependence models.[12] Then $|\mathcal{PA}^i| = \sum_{m=0}^{M} \binom{N}{m}$, $\sum_{s\in\mathcal{PA}^i} |s|^2 = \sum_{m=0}^{M} \binom{N}{m} m^2$, and $\sum_{s\in\mathcal{PA}^i} |s|^3 = \sum_{m=0}^{M} \binom{N}{m} m^3$, and Equation 3.50 can be written as:

$$
\begin{aligned}
&\Theta\left(\sum_{k=1}^{K}\sum_{i=1}^{N}\sum_{s\in\mathcal{PA}_k^i}\left((|s|RD_x)^2 + {D_y}^2\right)T_k + \left((|s|RD_x)^3 + {D_y}^3\right)\right) = \\
&\Theta\left(\sum_{i=1}^{N}\left[\left(\sum_{s\in\mathcal{PA}^i}|s|^2\right)(RD_x)^2 + |\mathcal{PA}^i|{D_y}^2\right]T + \left[\left(\sum_{s\in\mathcal{PA}^i}|s|^3\right)(RD_x)^3 + |\mathcal{PA}^i|{D_y}^3\right]K\right) \\
&\Theta\left(\left[(RD_x)^2\sum_{m=0}^{M}\binom{N}{m}m^2 + {D_y}^2\sum_{m=0}^{M}\binom{N}{m}\right]NT\right. \\
&\quad\left.+\left[(RD_x)^3\sum_{m=0}^{M}\binom{N}{m}m^3 + {D_y}^3\sum_{m=0}^{M}\binom{N}{m}\right]NK\right). \qquad (3.51)
\end{aligned}
$$

Unfortunately, there is no closed-form formula for expressions $\sum_{m=0}^{M} \binom{N}{m}$, $\sum_{m=0}^{M} \binom{N}{m} m^2$, and $\sum_{m=0}^{M} \binom{N}{m} m^3$. However, under some additional assumptions, these expressions can be simplified in the asymptotic analysis. For example, if $M \geq N/2$, then the following holds:

$$
\begin{aligned}
&\sum_{m=0}^{M}\binom{N}{m} \geq \sum_{m=0}^{\lceil N/2\rceil}\binom{N}{m} \geq \frac{2^N}{2} \\
&\sum_{m=0}^{M}\binom{N}{m}m^2 \geq \sum_{m=\lceil N/4\rceil}^{\lceil N/2\rceil}\binom{N}{m}m^2 \geq \frac{2^N}{4}\left(\frac{N}{4}\right)^2 = \frac{2^N N^2}{4^3} \qquad (3.52)\\
&\sum_{m=0}^{M}\binom{N}{m}m^3 \geq \sum_{m=\lceil N/4\rceil}^{\lceil N/2\rceil}\binom{N}{m}m^3 \geq \frac{2^N}{4}\left(\frac{N}{4}\right)^3 = \frac{2^N N^3}{4^4},
\end{aligned}
$$

as well as

$$
\begin{aligned}
&\sum_{m=0}^{M}\binom{N}{m} \leq \sum_{m=0}^{N}\binom{N}{m} \leq 2^N \\
&\sum_{m=0}^{M}\binom{N}{m}m^2 \leq \sum_{m=0}^{N}\binom{N}{m}m^2 \leq 2^N N^2 \qquad (3.53)\\
&\sum_{m=0}^{M}\binom{N}{m}m^3 \leq \sum_{m=0}^{N}\binom{N}{m}m^3 \leq 2^N N^3.
\end{aligned}
$$

[12]For simplicity, we assume here the same bound on the number of parents across all signals.

Combining the previous two sets of inequalities, it follows:

$$\begin{aligned}
\sum_{m=0}^{M} \binom{N}{m} &= \Theta(2^N) \\
\sum_{m=0}^{M} \binom{N}{m} m^2 &= \Theta(2^N N^2) \\
\sum_{m=0}^{M} \binom{N}{m} m^3 &= \Theta(2^N N^3) .
\end{aligned} \tag{3.54}$$

After plugging this result into Equation 3.51, we obtain

$$\begin{aligned}
&\Theta\left(\left[(RD_x)^2 \sum_{m=0}^{M} \binom{N}{m} m^2 + {D_y}^2 \sum_{m=0}^{M} \binom{N}{m}\right] NT \right. \\
&\left. + \left[(RD_x)^3 \sum_{m=0}^{M} \binom{N}{m} m^3 + {D_y}^3 \sum_{m=0}^{M} \binom{N}{m}\right] NK\right) \\
&= \Theta\left(\left[(RD_x)^2 2^N N^2 + {D_y}^2 2^N\right] NT + \left[(RD_x)^3 2^N N^3 + {D_y}^3 2^N\right] NK\right) \\
&= \Theta\left(\left[(NRD_x)^2 + {D_y}^2\right] 2^N NT + \left[(NRD_x)^3 + {D_y}^3\right] 2^N NK\right) .
\end{aligned} \tag{3.55}$$

Note that this is exactly the same result as the one in Equation 3.49, which holds whenever at least a constant fraction of subsets is allowed (1/2 in this case).

If the number of allowed parents, M, is small, a better (lower) time complexity could be achieved. In many scenarios, M does not depend on the number of signals, N. For example, it is reasonable to assume that there is a limit on how many people one person can simultaneously react to, which is independent on the number of people in a scene, and so the same bound can be used whether there is only a handful of people or a large crowd in it. In such scenarios, M can be treated as a constant, and the expressions $\sum_{m=0}^{M} \binom{N}{m}$, $\sum_{m=0}^{M} \binom{N}{m} m^2$, and $\sum_{m=0}^{M} \binom{N}{m} m^3$ all have the complexity $\Theta(N^M)$ ($N(N-1)\ldots(N-M+1)$ is an M^{th}-order polynomial in N, and the summation is over M terms, which is a constant number). Now, Equation 3.51 can be reduced to:

$$\begin{aligned}
&\Theta\left(\left[(RD_x)^2 \sum_{m=0}^{M} \binom{N}{m} m^2 + {D_y}^2 \sum_{m=0}^{M} \binom{N}{m}\right] NT \right. \\
&\left. + \left[(RD_x)^3 \sum_{m=0}^{M} \binom{N}{m} m^3 + {D_y}^3 \sum_{m=0}^{M} \binom{N}{m}\right] NK\right) \\
&= \Theta\left(\left[(RD_x)^2 N^M + {D_y}^2 N^M\right] NT + \left[(RD_x)^3 N^M + {D_y}^3 N^M\right] NK\right) \\
&= \Theta\left(\left[(RD_x)^2 + {D_y}^2\right] N^{M+1} T + \left[(RD_x)^3 + {D_y}^3\right] N^{M+1} K\right) .
\end{aligned} \tag{3.56}$$

On the other hand, if M is not treated as a constant, or simply a more precise statement about time complexity is needed – one that describes the dependence on M as well (whether a constant or not), then a simple bound can be obtained as:

$$\begin{aligned}
&\Theta\left(\left[(RD_x)^2\sum_{m=0}^{M}\binom{N}{m}m^2 + {D_y}^2\sum_{m=0}^{M}\binom{N}{m}\right]NT\right.\\
&\left.+\left[(RD_x)^3\sum_{m=0}^{M}\binom{N}{m}m^3 + {D_y}^3\sum_{m=0}^{M}\binom{N}{m}\right]NK\right)\\
&=\mathcal{O}\left(\left[(RD_x)^2\binom{N}{M}M^3 + {D_y}^2\binom{N}{M}M\right]NT + \left[(RD_x)^3\binom{N}{M}M^4 + {D_y}^3\binom{N}{M}M\right]NK\right)\\
&=\mathcal{O}\left(\left[(MRD_x)^2 + {D_y}^2\right]\binom{N}{M}MNT + \left[(MRD_x)^3 + {D_y}^3\right]\binom{N}{M}MNK\right). \qquad (3.57)
\end{aligned}$$

Here, we used the fact that $\binom{N}{0} < \binom{N}{1} < \ldots < \binom{N}{M-1} < \binom{N}{M}$ for $M \leq N/2$, and, consequently, that $\sum_{m=0}^{M}\binom{N}{m} \leq M\binom{N}{M}$. A better bound can in fact be obtained under the assumption that $M/N \leq c$, where c is a constant smaller than $1/2$. In other words, we now allow that M grows as N grows, as long as M/N does not grow. Under this assumption, it holds that

$$\frac{\binom{N}{m-1}}{\binom{N}{m}} = \frac{m}{N-m+1} \leq \frac{M}{N-M+1} \leq \frac{cN}{N-cN+1} \leq \frac{c}{1-c} < 1 \qquad (3.58)$$

for $1 \leq m \leq M$. Let $c' = c/(1-c)$. It follows that

$$\binom{N}{m} \leq {c'}^{N-m}\binom{N}{M} \qquad (3.59)$$

and

$$\sum_{m=0}^{M}\binom{N}{m} \leq \binom{N}{M}\sum_{m=0}^{M}{c'}^{N-m} \leq \binom{N}{M}\sum_{m=0}^{\infty}{c'}^{N-m} = \frac{1}{1-c'}\binom{N}{M}, \qquad (3.60)$$

where the last equality follows from $c' < 1$. Similarly,

$$\begin{aligned}
\sum_{m=0}^{M}\binom{N}{m}m^2 &\leq \frac{1}{1-c'}\binom{N}{M}M^2\\
\sum_{m=0}^{M}\binom{N}{m}m^3 &\leq \frac{1}{1-c'}\binom{N}{M}M^3.
\end{aligned} \qquad (3.61)$$

On the other hand, the three sums are also bounded below as

$$\begin{aligned}
\sum_{m=0}^{M} \binom{N}{m} &\geq \binom{N}{M} \\
\sum_{m=0}^{M} \binom{N}{m} m^2 &\geq \binom{N}{M} M^2 \\
\sum_{m=0}^{M} \binom{N}{m} m^3 &\geq \binom{N}{M} M^3 .
\end{aligned} \tag{3.62}$$

Finally, from Equations 3.60, 3.61 and 3.62, it follows that

$$\begin{aligned}
\sum_{m=0}^{M} \binom{N}{m} &= \Theta\left(\binom{N}{M}\right) \\
\sum_{m=0}^{M} \binom{N}{m} m^2 &= \Theta\left(\binom{N}{M} M^2\right) \\
\sum_{m=0}^{M} \binom{N}{m} m^3 &= \Theta\left(\binom{N}{M} M^3\right) ,
\end{aligned} \tag{3.63}$$

which, when plugged into Equation 3.51, yields the time complexity of

$$\begin{aligned}
&\Theta\left(\left[(RD_x)^2 \sum_{m=0}^{M} \binom{N}{m} m^2 + {D_y}^2 \sum_{m=0}^{M} \binom{N}{m}\right] NT \right. \\
&\quad \left. + \left[(RD_x)^3 \sum_{m=0}^{M} \binom{N}{m} m^3 + {D_y}^3 \sum_{m=0}^{M} \binom{N}{m}\right] NK\right) \\
&= \Theta\left(\left[(RD_x)^2 \binom{N}{M} M^2 + {D_y}^2 \binom{N}{M}\right] NT + \left[(RD_x)^3 \binom{N}{M} M^3 + {D_y}^3 \binom{N}{M}\right] NK\right) \\
&= \Theta\left(\left[(MRD_x)^2 + {D_y}^2\right] \binom{N}{M} NT + \left[(MRD_x)^3 + {D_y}^3\right] \binom{N}{M} NK\right) .
\end{aligned} \tag{3.64}$$

Note that this is a tighter bound than the one in Equation 3.57 by a factor of M, due to a more precise analysis of the complexity of the sum of binomial coefficients when M/N is bounded, which also encompasses the case of small (constant) M.

In Algorithm 2.4, for every model and every signal, updated hyperparameters are computed for each allowed parent set. These hyparameters are only used here to compute the marginal data likelihood and update the prior parameter on the parent set. To minimize memory requirements, these hyparameters can be discarded, and only the updated parameter of the parent set prior, ${\beta'}_{i,s}^{k}$ can be kept for each allowed parent set, which takes $\Theta\left(|\mathcal{PA}_k^i|\right)$ space. However, after a parent set is sampled, these values are

not needed any more, and so the total memory requirement for storing "beta" values is $\Theta\left(\max_{k,i}|\mathcal{PA}_k^i|\right) = \mathcal{O}\left(2^N\right)$. In the case of a bounded-indegree prior, this is equal to $\Theta\left(2^N\right)$ if $M \geq N/2$ and to $\Theta\left(\binom{N}{M}\right)$ if $M/N \leq c < 1/2$. In addition, a sample of a parent set, $\tilde{pa}(i,k)$, and parameters of the dependence model, $\tilde{A}_k^i$ and $\tilde{Q}_k^i$, is stored for every model k and every signal i. The size of $\tilde{pa}(i,k)$ is N in the worst case. If a bounded-indegree prior is employed, it is M in the worst case. The average case analysis depends on the particular value of the prior and the data. Here, we will assume that the average size of a parent set is $\Theta(M)$. The dimensions of parameters $\tilde{A}_k^i$ and $\tilde{Q}_k^i$ are $D_y \times |\tilde{pa}(i,k)|RD_x$ and $D_y \times D_y$, respectively. Thus, the total memory requirement for storing samples is $\mathcal{O}\left(ND_y(NRD_x + D_y)K\right)$ in general, and $\mathcal{O}\left(ND_y(MRD_x + D_y)K\right)$ if a bounded-indegree prior is used. Finally, the overall memory complexity of Algorithm ?? is $\mathcal{O}\left(2^N + ND_y(NRD_x + D_y)K\right)$ in general and $\Theta\left(\binom{N}{M} + ND_y(MRD_x + D_y)K\right)$ in case of a bounded indegree prior in which $M/N \leq c < 1/2$.

It is important to note that in our implementation, posterior over parent sets and dependence model parameters is computed and stored for all models and signals. Storing parameters of these posteriors takes $\Theta\left(\sum_{k=1}^{K}\sum_{i=1}^{N}\sum_{s\in\mathcal{PA}_k^i}\left((|s|RD_x)^2 + {D_y}^2\right)\right)$ space, which is in general bounded by $\mathcal{O}\left(\left((NRD_x)^2 + {D_y}^2\right)2^N NK\right)$ and is equal to $\Theta\left(\left((MRD_x)^2 + {D_y}^2\right)\binom{N}{M}NK\right)$ in case of a bounded-indegree prior with small M or if $M/N \leq c < 1/2$ is satisfied. Although this significantly increases the space required for this step, it does not increase the overall space complexity of the inference procedure, since the same space is required for storing parameters of prior distributions in general.[13]

■ 3.5.5 Complexity of inference in LG-SSIM: step 5

Algorithm 3.2 describes a procedure for sampling the observation noise covariance matrix in LG-SSIM that assumes the same observation model for all signals and all time points. For each time point, t, and each signal, i, the value $(Y_t^i - C'^0 X_t'^i)(Y_t^i - C'^0 X_t'^i)^T$ is a statistic that must be computed in order to update the inverse-Wishart prior on the observation noise covariance matrix. Here, Y_t^i is a vector of length D_y, $X_t'^i$ is a vector of length RD_x, and C'^0 is a matrix of dimension $D_y \times RD_x$. Therefore, computing $C'^0 X_t'^i$ takes $\Theta(D_y RD_x)$ time and evaluating the product $(Y_t^i - C'^0 X_t'^i)(Y_t^i - C'^0 X_t'^i)^T$ takes $\Theta(D_y^2)$ time. These are computationally dominant steps, and thus, updating the prior takes $\Theta\left(D_y(RD_x + D_y)NT\right)$ time in total. That can also be considered the overall complexity of Algorithm 3.2, since generating a sample from the inverse-Wishart distribution takes $\Theta(D_y^3)$ time[14] and does not depend on the number of time points T,

[13] In general, prior on dependence model parameters can be set independently for each parent set of each signal in each switching model. However, if these priors are constructed in some parametric way, thy may be represented more compactly.

[14] The implementation of an algorithm for sampling from an inverse-Wishart distribution that we are using performs matrix operations that are cubic in time (such as QR-decomposition).

which is typically much larger than D_y.

Note that, in general, we assume that an observation of a signal, Y_t^i, is a function of the corresponding expanded state, X'^i_t, which means that it is a function of a signal value over some window of time in the past. However, in most practical scenarios, it will be the case that an observation of a signal is only a function of its current state, X_t^i. With that assumption, an equivalent computation $(Y_t^i - C^0 X_t^i)(Y_t^i - C^0 X_t^i)^T$ can be used instead, where matrix C^0 is of dimension $D_y \times D_x$, which would reduce the total time complexity of this step to $\Theta\left(D_y(D_x + D_y)NT\right)$. However, this is not critical for the performance of the overall Gibbs sampling procedure (Algorithm 3.1) as this step is far less computationally demanding than steps 1 and 2 even without such optimization.

Algorithm 3.2 does not require significant additional space, except for storing the value of an updated hyparameter Ψ'^0_R of dimension $D_y \times D_y$, as well as matrices of the same size during the sampling substep. However, for convenience, our implementation creates copies of the state and observation sequences in a "reshaped" format convenient for applying matrix operations in MATLAB, and therefore takes $\Theta\left(D_y^2 + (RD_x + D_y)NT\right)$ of additional space. Still, that is of the same memory complexity as the input to this step, and significantly cheaper than the space required for step 1, and thus not critical.

Chapter 4

SSIM Experiments

THE main goal of this thesis is to develop tools for learning time-varying interactions among signals from noisy observations of these signals. In doing that, we employ a Bayesian approach that characterizes uncertainty of latent variables via their posterior distribution. The goal of this Chapter is twofold. First, it aims at illustrating a variety of analyses that could be performed and questions that could be answered (probabilistically) using the SSIM framework. Second, it demonstrates the advantage of the SSIM model over the previous work by comparing results of interaction analysis obtained by the SSIM with the ones obtained by the model that does not account for noisy observations [49, 50].

We present experimental results on three datasets: synthetic data, joystick data, and climate data. **Synthetic data** is generated to demonstrate specific advantages of the SSIM model. Most real data does not contain annotation of interactions. Furthermore, ground truth interactions are in most cases hard to label even by domain experts. **Joystick data** is generated by humans in an experiment that is specifically devised for testing the SSIM inference algorithm in a realistic scenario. In this experiment, players control a point on a screen via joystick in such a way that they interact with only a predetermined subset of players in a specific way. Patterns of interaction change over time also by a predetermined schedule. Thus, joystick data contains ground truth interactions and switching pattern by design, and is therefore suitable for testing interaction analysis. Finally, **climate data** is a real-world data of historical values of different climate indices that cover various aspects of climate. It is still largely unknown how climate exactly works and uncovering relationships among climate indices is one of the tasks that may contribute towards its understanding. As the ground truth is not known, this dataset is mainly used to demonstrate the variety of applications and types of analyses enabled by the methodology developed in this thesis.

Note that in this thesis we focus on continuous-valued time-series data, in which inference can be done using the LG-SSIM model. This is indeed the case with the three datasets used in this Chapter.

In addition, there are practical considerations that are critical to address for a successful employment of the Gibbs sampling procedure for LG-SSIM: setting hyperparameters, initializing latent variables, choosing a Gibbs sampling scehdule, and extracting statistics from the posterior samples. We discuss these first and then present

experimental results.

In Section 4.1, we provide guidelines for setting the prior (i.e., hyperparameters) in the LG-SSIM model, initializing latent variables, and performing a Gibbs sampling procedure. We also provide a procedure for evaluating a posterior distribution over a huge number of structures given a limited (much smaller) number of posterior samples obtained by the Gibbs sampling inference procedure. In Sections 4.2, we use synthetic data to demonstrate the advantage of interaction analysis over testing pairwise relationships, and the advantage of the SSIM model over the model of Siracusa and Fisher [49, 50], which does not account for observation noise. In Section 4.3, we introduce a novel dataset, the joystick data, which is created specifically for testing results of interaction analysis in realistic conditions. It is developed in such a way that ground truth interactions are known by design, but it is human-generated and not synthesized from the model. We demonstrate the ability of the SSIM model to infer interactions and a switching pattern even in the presence of relatively high observation noise or if a significant fraction of data is missing, and that it is advantageous over the STIM model of Siracusa and Fisher [49, 50], as the STIM model does not handle missing data and performs worse in the presence of high observation noise. We also demonstrate the advantage of reasoning over structure posterior over MAP estimation, as spurious edges in a MAP structure estimate are typically assigned higher uncertainty (lower probability) in the posterior than the correct edges. Finally, in Section 4.4, we apply the SSIM model to a real-world problem and show types of analyses that it enables.

■ 4.1 Implementation and Practical Considerations

Inference in LG-SSIM (and SSIM in general) is inherently hard. Since exact inference is intractable, we employ a Gibbs sampling procedure described in 3.4 for approximate inference. However, although Gibbs sampling has a theoretical guarantee that the obtained samples will converge to the correct posterior distribution, obtaining a representative set of samples from the posterior in limited time is challenging in LG-SSIM. The space of latent variables and model parameters is very complex. The posterior distribution is highly multimodal, and there may be many local optima, and, as a result, the sampling algorithm may easily get stuck in a wrong subspace of solutions.

Here, we discuss practical considerations that need to be addressed in order to successfully employ LG-SSIM. First, the results of inference can be very sensitive to the value of the parameters of the prior (i.e., hyperparameters) and the initial values of latent variables. The values of hyperparameters directly bias the posterior distribution, especially when only a limited data is available, which is a regime of particular interest in this thesis. Therefore, the closer the prior is to the "truth", the better the results will be. In order to set the prior as good as possible, we use common sense, prior knowledge, as well as data itself as a guide. Furthermore, setting the values of hyperparameters and initial values of latent variables properly is instrumental in focusing the Gibbs sampler into a region of interest. This is of critical importance since the posterior

distribution under the SSIM model is highly multimodal due to a very high complexity of the latent space. Also, the exact sampling schedule (the order of steps, the burn-in period, the distance between samples taken from a chain, and the number of restarts) plays a critical role in efficiently traversing the posterior space and generating valid samples from the posterior distribution. Finally, due to a huge number of structures to reason over, the number of posterior samples that can be generated in reasonable time is typically much smaller than that, and therefore only a small fraction of structures would be assigned a non-zero posterior probability. In order to overcome such sparsity and obtain a more precise posterior picture, we modify the posterior analysis in such a way that conditioned on each joint sample of other latent variables in the model, a full probability distribution over structures is constructed, and the final result is obtained by averaging over these distributions.

■ 4.1.1 Setting Up The Prior

Prior on LG-SSIM can be thought of as a collection of priors on different parts of the model. Here, we analyze how each of them may influence results of inference and provide guidelines on how to set hyperparameters.

Prior on switching model

The switching model consists of $K+1$ multinomial distribution, $\mathcal{M}ult(\pi_1, \ldots, \pi_K)$ and $\mathcal{M}ult(\pi_{k,1}, \ldots, \pi_{k,K})$, $k = 1, \ldots, K$, that govern the evolution of the switching sequence:

$$\begin{aligned} Z_1 &\sim \mathcal{M}ult(\pi_1, \ldots, \pi_K) \\ Z_t &\sim \mathcal{M}ult(\pi_{Z_{t-1},1}, \ldots, \pi_{Z_{t-1},K}), \quad t = 2, \ldots, T. \end{aligned} \tag{4.1}$$

The prior on the switching model consists of $K+1$ Dirchlet distributions that are priors to the corresponding multinomials:

$$\begin{aligned} (\pi_1, \ldots, \pi_K) &\sim \mathcal{D}ir(\alpha_1, \ldots, \alpha_K) \\ (\pi_{k,1}, \ldots, \pi_{k,K}) &\sim \mathcal{D}ir(\alpha_{k,1}, \ldots, \alpha_{k,K}), \quad k = 1, \ldots, K. \end{aligned} \tag{4.2}$$

Recall that the mean of this prior is:

$$\begin{aligned} \mathrm{E}_{prior}\left[(\pi_1, \ldots, \pi_K)\right] &= (\alpha_1, \ldots, \alpha_K) \Big/ \sum_{k=1}^{K} \alpha_k \\ \mathrm{E}_{prior}\left[(\pi_{k,1}, \ldots, \pi_{k,K})\right] &= (\alpha_{k,1}, \ldots, \alpha_{k,K}) \Big/ \sum_{k'=1}^{K} \alpha_{k,k'}, \quad k = 1, \ldots, K. \end{aligned} \tag{4.3}$$

Therefore, the prior on the switching model introduces a bias towards these values of initial and transition probabilities. The strength of this bias is controlled by the

variance , which is

$$\begin{aligned} \mathrm{Var}_{prior}\left[\pi_k\right] &= \frac{\alpha_k(\alpha_0-\alpha_k)}{\alpha_0^2(\alpha_0+1)}, \quad k=1,\ldots,K \\ \mathrm{Var}_{prior}\left[\pi_{k,k'}\right] &= \frac{\alpha_{k,k'}(\alpha_{k,0}-\alpha_{k,k'})}{\alpha_{k,0}^2(\alpha_{k,0}+1)}, \quad k,k'=1,\ldots,K, \end{aligned} \tag{4.4}$$

where $\alpha_0 = \sum_{k=1}^{K} \alpha_k$ and $\alpha_{k,0} = \sum_{k'=1}^{K} \alpha_{k,k'}$. Note that the variance decreases with the sum of hyperparameters, and consequently, the strength of the prior increases. In conclusion, the parameters $\alpha_{k,1},\ldots,\alpha_{k,K}$ are proportional to the expected prior transition probabilities, while their sum, $\alpha_{k,0}$, controls the strength of the prior. These parameters are called pseudocounts, as increasing $\alpha_{k,k'}$ by an integer value has the same effect on the posterior as if there were that many additional observed transitions from k to k'.

We treat all states equally in the prior. In most applications, self-transitions are much more likely than transitions to other states. Note that $\pi_{k,k}/(1-\pi_{k,k}) = \alpha_{k,k}/(\sum_{k'\neq k}\alpha_{k,k'})$. Thus, to set the prior such that from state k self-transition is m times more likely than a transition to another state, we set $\alpha_{k,k'} = \alpha_{k,k}/(m(K-1))$ for $k' \neq k$. Unless there is prior knowledge of frequent switching, we set m to be $T'/(K-1)$. In other words, the prior expectation is to see approximately $K-1$ transitions within the sequence of length T'. Therefore, $\alpha_{k,k'} = \alpha_{k,k}/T'$. For example, $\alpha_{k,k'} = 1$ for $k' \neq k$ and $\alpha_{k,k} = T'$ is a common setting that we use, for some $T' \geq 100$. Note that this implies that for relatively short sequences the pseudo-count is on the order of the length of the sequence. While that is a moderately strong prior on switching parameters, note that the posterior of the switching sequence Z is heavily influenced by the observed time-series, and not just switching parameters. In addition, we set $\alpha_k = 1$ for $k = 1,\ldots,K$. Note that this prior is not very influential – initial state is mostly driven by the data. If there is only one sequence, as is the case in many applications we consider, this prior is not important.

Prior on dependence models

Each of the K dependence models consists of a dependence graph, $\tilde{E}_k$, and a set of linear Gaussian models,

$$X_t^i = \tilde{A}_k^i X_{t-1}^{\tilde{pa}(i,k)} + w_t^i, \qquad w_t^i \sim \mathcal{N}(0, \tilde{Q}_k^i), \tag{4.5}$$

one for each signal $i = 1,\ldots,N$, parametrized by the dependence matrix $\tilde{A}_k^i$ and noise covariance matrix $\tilde{Q}_k^i$. The prior on k^{th} dependence model can be written as

$$p(\tilde{E}_k, \tilde{\theta}_k; \beta^k, \gamma^k) = p(\tilde{E}_k; \beta^k)\, p(\tilde{\theta}_k \mid \tilde{E}_k; \gamma^k), \tag{4.6}$$

where $\tilde{\theta}_k = \{(\tilde{A}_k^i, \tilde{Q}_k^i)\}_{i=1}^N$ in the case of LG-SSIM. Since we use a modular prior, it can be decomposed as a product of priors on parent sets and parameters associated with

each signal's evolution model:

$$p(\tilde{E}_k, \tilde{\theta}_k; \beta^k, \gamma^k) = \prod_{i=1}^{N} p(\tilde{pa}(i,k); \beta^k)\; p(\tilde{\theta}_k^i \mid \tilde{pa}(i,k); \gamma^k)\;. \tag{4.7}$$

The **prior on a parent set** of signal i has a general form:

$$p(\tilde{pa}(i,k); \beta^k) = \frac{1}{B_i^k} \beta_{i,\tilde{pa}(i,k)}^k\;, \tag{4.8}$$

where $B_i^k = \sum_{\tilde{pa}(i,k)} \beta_{i,\tilde{pa}(i,k)}^k$ is a normalization constant. Since the set of possible parent sets of a signal is large (2^N), it is critical, even for moderate N, to work with a subset of manageable size. Domain knowledge should be utilized to consider only a fraction of parent sets that are most likely a priori. Excluded parent sets can be treated as having prior probability 0. Ideally, if excluded parent sets are unlikely, that is an accurate reflection of the prior knowledge. However, if that is not the case, but these parent sets must be excluded for practical reasons, thus obtained model can be thought of as a tractable approximation to the real world. Alternatively, the dependence model of an individual signal can be thought of as a mixture model, where the mixture is over a selected subset of parent sets.

Let S_i^k be the set of allowed parent sets of signal i in the k^{th} model. We assume the following form of the prior on its parent set:

$$\beta_{i,s}^k = \begin{cases} \dfrac{1}{(|s|+1)^{b_{k,i}}} & , s \in S_i^k \\ 0 & , \text{o.w.} \end{cases} \tag{4.9}$$

In other words, the prior probability of a parent set is inversely proportional to the size of the parent set (plus one, to accommodate an empty set), raised to an exponent. If $b_{k,i} > 0$, the prior favors smaller parent sets. If $b_{k,i} < 0$, the prior favors larger parent sets. Finally, If $b_{k,i} = 0$, the prior probability of all parent sets is equal. Note that when $b_{k,i} > 0$, the prior acts as a regularization term on the number of parents.

One would be tempted to conclude that it is critical to use such a prior, which penalizes large parent sets, as in the case with AIC and BIC model selection criteria. However, that is not necessarily the case in SSIM since parameters are marginalized out to compute the posterior distribution of a parent set. Marginalization of parameters has the effect of averaging data likelihood over all parameter values, weighted by the prior on parameters. A larger parent set results in larger number parameters, but that does not mean that averaging over a larger set of parameters would yield a higher likelihood. In fact, if an additional parameter is not relevant (i.e., if the best model that includes it is not significantly better than the best model without it), then likely most of its values would contribute to the decrease of the average likelihood. Of course, that also depends on the prior on parameters, due to weighting. The exact relationship between the prior

and the average likelihood (obtained by marginalizing parameters) is complicated and we do not investigate it here (it would be an important future work).

On the other hand, if a joint MAP estimation of a parent set and parameters is performed, then some form of regularization is needed – either via a prior on parent set that penalizes large sets ($b_{k,i} > 0$), or via a prior on parameters that promotes sparsity (L_1 penalty may be needed instead of L_2).

We typically use $b_{k,i} \geq 0$, and commonly $1 \leq b_{k,i} \leq 10$ (default being $b_{k,i} = 1$), so that there is no high bias for smaller parent sets, and the posterior is mostly guided by the data and parameter averaging, but still favors smaller sets in order to provide regularization when data size is small. Since the SSIM model is only an approximation to the true process of interest in a particular application (time discretization and the assumption of a linear Gaussian transition model), the MAP parent set may not reflect the causal structure even with unlimited data. $b_{k,i}$ may be set significantly higher (or progressively increasing in repeated experiments) to "prune" the parent sets further. This can be done in an exploratory analysis in an attempt to uncover possible causal structures. When exactly this would be possible or beneficial requires further investigation.

Also, we typically constrain the set of parent sets by assuming that a signal is always included in in its parent set (which is true in most applications), and that there can be at most M parents (bounded-indegree prior), implying $S_i^k = \{s \mid i \in s, |s| \geq M\}$.

The **prior on parameters** associated with i^{th} signal evolution model, $\tilde{A}_k^i$ and $\tilde{Q}_k^i$, is a matrix normal inverse-Wishart distribution:

$$\begin{aligned} & p(\tilde{A}_k^i, \tilde{Q}_k^i \mid \tilde{pa}(i,k); M_k^{i,\tilde{pa}(i,k)}, \Omega_k^{i,\tilde{pa}(i,k)}, \kappa_k^{i,\tilde{pa}(i,k)}, \Psi_k^{i,\tilde{pa}(i,k)}) \\ & = \mathcal{MN}(\tilde{A}_k^i; M_k^{i,\tilde{pa}(i,k)}, \Omega_k^{i,\tilde{pa}(i,k)}, \tilde{Q}_k^i)\, \mathcal{IW}(\tilde{Q}_k^i; \kappa_k^{i,\tilde{pa}(i,k)}, \Psi_k^{i,\tilde{pa}(i,k)}) . \end{aligned} \tag{4.10}$$

Note that this prior is conditioned on the parent set, $\tilde{pa}(i,k)$. Therefore, for each possible value of the parent set, there are separate hyperparameters, $\kappa_k^{i,\tilde{pa}(i,k)}$, $\Psi_k^{i,\tilde{pa}(i,k)}$, $M_k^{i,\tilde{pa}(i,k)}$ and $\Omega_k^{i,\tilde{pa}(i,k)}$. We observe that the results of the Gibbs sampling inference procedure for LG-SSIM are very sensitive to these hyperparameters and we pay particular attention to setting them appropriately.

Since the prior on the transition matrix, $\tilde{A}_k^i$, is conditioned on the noise covariance matrix, $\tilde{Q}_k^i$, it is natural to consider first the prior on $\tilde{Q}_k^i$. Recall that in the inverse-Wishart prior, $\kappa_k^{i,\tilde{pa}(i,k)}$ has a role of a pseudocount, while $\Psi_k^{i,\tilde{pa}(i,k)}$ is proportional to the mean of $\tilde{Q}_k^i$:

$$\mathrm{E}_{prior}\left[\tilde{Q}_k^i \mid \tilde{pa}(i,k)\right] = \frac{\Psi_k^{i,\tilde{pa}(i,k)}}{\kappa_k^{i,\tilde{pa}(i,k)} - d_i - 1}, \tag{4.11}$$

where d_i is the number of rows / columns of $\tilde{Q}_k^i$, and does not depend on the parent set. Pseudocount, $\kappa_k^{i,\tilde{pa}(i,k)}$, has the effect as if there were that many samples of the covariance matrix whose sum equals $\Psi_k^{i,\tilde{pa}(i,k)}$ (or, are $\Psi_k^{i,\tilde{pa}(i,k)} / \kappa_k^{i,\tilde{pa}(i,k)}$ on average).

We typically set $\kappa_k^{i,\tilde{pa}(i,k)}$ small, which implies a weak prior on $\tilde{Q}_k^i$. Specifically, in most experiments we use $\kappa_k^{i,\tilde{pa}(i,k)} = d_i + 3$, which is only slightly higher than $d_i - 1$, the minimum allowed value for parameter $\kappa_k^{i,\tilde{pa}(i,k)}$. We set $\Psi_k^{i,\tilde{pa}(i,k)}$ in the following way. The noise covariance of signal i is estimated from data using a simpler model that is a vector autoregressive model of signal i:

$$X_t^i = A_{ind}^i X'^i_{t-1} + w_t^i, \quad w_t \sim \mathcal{N}(0, Q_{ind}^i) , \tag{4.12}$$

where $X'^i_{t-1} = \left[{X_{t-1}^i}^T \dots {X_{t-r}^i}^T\right]^T$. Note that this model does not take into account switching, observation noise, as well as other signals, and therefore estimation can be done independent of the prior on other parts of the LG-SSIM. The maximum likelihood estimates of A_{ind}^i and Q_{ind}^i are computed as

$$\begin{aligned} \hat{A}_{ind}^i &= \left(\sum_{t\in\mathcal{T}_{obs}} X_t^i {X'^i_{t-1}}^T\right)\left(\sum_{t\in\mathcal{T}_{obs}} X'^i_{t-1} {X'^i_{t-1}}^T\right)^{-1} \\ \hat{Q}_{ind}^i &= \frac{1}{|\mathcal{T}_{obs}|}\sum_{t\in\mathcal{T}_{obs}} (X_t^i - \hat{A}_{ind}^i X'^i_{t-1})(X_t^i - \hat{A}_{ind}^i X'^i_{t-1})^T , \end{aligned} \tag{4.13}$$

where $\mathcal{T}_{obs}$ is a set of time indices for which observations of both X_t^i and X'^i_{t-1} exist. Alternatively, some of the missing values can be added, e.g., by interpolation, to extend $\mathcal{T}_{obs}$. That may be particularly important if r is large and the frequency of missing data is large. The estimated driving noise variance of signal i components (diagonal terms of $\hat{Q}_{ind}^i$) can be thought of as an upper bound to the corresponding variance in any of the K dependence models in LG-SSIM (diagonal terms of $\tilde{Q}_k^i$), since by adding other signals, allowing switching, and modeling observation process necessarily result in a model that fits the data better. Finally, $\Psi_k^{i,\tilde{pa}(i,k)}$ is set as such that the mean of the inverse-Wishart prior on $\tilde{Q}_k^i$ (Equation 4.11) is equal to $\hat{Q}_{ind}^i$, i.e.,

$$\Psi_k^{i,\tilde{pa}(i,k)} = \hat{Q}_{ind}^i (\kappa_k^{i,\tilde{pa}(i,k)} - d_i - 1) . \tag{4.14}$$

Now, we discuss setting the prior on the transition matrix, $\tilde{A}_k^i$. Recall that it is conditioned on $\tilde{Q}_k^i$, and has the form

$$p(\tilde{A}_k^i \,|\, \tilde{pa}(i,k), \tilde{Q}_k^i; M_k^{i,\tilde{pa}(i,k)}, \Omega_k^{i,\tilde{pa}(i,k)}) = \mathcal{MN}(\tilde{A}_k^i; M_k^{i,\tilde{pa}(i,k)}, \Omega_k^{i,\tilde{pa}(i,k)}, \tilde{Q}_k^i) . \tag{4.15}$$

The transition matrix is not know in advance, and we want to set its prior to be close to a uniform distribution. To achieve that, we set the column covariance matrix parameter, $\Omega_k^{i,\tilde{pa}(i,k)}$, such that its diagonal values are very high (e.g., 10^4 divided by the average diagonal element of $\hat{Q}_{ind}^i$, such that the variance of each element of $\tilde{A}_k^i$ is approximately

equal to 10^4 on average).[1] Note that this setting is very different from other works (e.g., [?]), in which $\Omega_k^{i,\tilde{pa}(i,k)}$ is set to identity. The reason is that in these works there is no inference on parent sets (a signal is influenced by all other signals), and the transition matrix is regularized by a prior that encourages small values. Here, model selection is performed via a posterior distribution on parent sets, while the transition matrix, $\tilde{A}_k^i$, is allowed to be arbitrary. In fact, such a setting is essential in LG-SSIM for proper inference of parent sets, since regularizing the transition matrix may render some parent sets unlikely simply due to a constraint on the transition matrix, while allowing an arbitrary transition matrix may result in their high posterior probability. Finally, the mean of the matrix normal distribution, $M_k^{i,\tilde{pa}(i,k)}$, is set to zero, although it is less relevant due to the large width of the prior. Alternatively, elements of $M_k^{i,\tilde{pa}(i,k)}$ that correspond to self prediction of a signal may be set to $\hat{A}_{ind}^i$, which is their maximum likelihood estimate in the model of that signal individually (Equation 4.13).

Prior on the observation model

We assume that the observation model is shared across all signals and is given by (Equation 3.12):

$$Y_t^i = C^0 X_t^i + v_t^i, \qquad v_t^i \sim \mathcal{N}(0, R^0). \tag{4.16}$$

We also assume that the observation matrix C^0 is known and equal to identity, and that the prior on the covariance matrix R^0 is the inverse-Wishart conjugate prior (Equation 3.13):

$$p(R^0; \kappa_R^0, \Psi_R^0) = \mathcal{IW}(R^0; \kappa_R^0, \Psi_R^0). \tag{4.17}$$

We typically set κ_R^0 small (e.g., $\kappa_R^0 = d + 2$, where d is the dimension of each signal), which implies a weak prior on R^0. The mean of the prior is set to be smaller than the the one on the dependence model noise (on average, over all signals). We typically set

$$\Psi_R^0 = 0.75 \frac{1}{N} \sum_{i=1}^{N} \mathrm{E}\left[\mathrm{diag}(\hat{Q}_{ind}^i)\right] (\kappa_R^0 - d - 1) I_{Nd \times Nd}, \tag{4.18}$$

where $\frac{1}{N}\sum_{i=1}^{N} \mathrm{E}\left[\mathrm{diag}(\hat{Q}_{ind}^i)\right]$ is the average value of the upper bound on variance across all variables in all signals (recall that $\hat{Q}_{ind}^i$) is estimated using Equation 4.13), and $I_{Nd \times Nd}$ is an identity matrix of dimension $Nd \times Nd$. The diagonal terms of the prior mean on the observation noise covariance (terms that correspond to variances of individual signal variables) are set slightly lower than the corresponding terms of the prior mean on the dependence noise covariance in order to prevent explaining the data just with a high observation noise.

[1] The matrix normal distribution of Equation 4.15 is equivalent to a multivariate normal distribution on $\mathrm{vec}(\tilde{A}_k^i)$ with covariance matrix $\Omega_k^{i,\tilde{pa}(i,k)} \oplus \tilde{Q}_k^i$.

■ 4.1.2 Setting up the Gibbs Sampler

A Gibbs sampling algorithm is guaranteed to converge to the correct posterior distribution. However, the convergence may be very slow, especially if a distribution has multiple local maxima. Since samples are typically highly correlated with the previous samples, it may take a very long time for the sampler to "escape" one local maximum. The resulting samples can therefore depend significantly on the initial value of latent variables and parameters, as well as on which samples are taken from a Markov chain and the number of times the chain is reinitialzied.

Initializing Latent Variables

Initial values of latent variables and parameters in the SSIM can significantly influence the distribution of the samples obtained in a reasonable time due to the high complexity of the latent space. Guessing the K dependence models a priori is difficult, as there is typically no evidence of what they should be (and inferring their interaction structures is in fact the main goal of the thesis). Setting them randomly may bias the algorithm towards wrong explanations of the data. Therefore, we initialize other variables first and then sample dependence models conditioned on other variables, as in the Gibbs sampling procedure. Guessing the switching sequence, Z, is also difficult (unless there is some strong prior knowledge). We typically initialize it randomly, such that the value of the switching sequence at each time point is drawn independently from a uniform distribution over the possible switching states. That avoids the bias towards any particular pattern, as well as bias towards self-transitions, which helps make larger moves through the posterior in the initial rounds of the sampler. We initialize the latent time-series state sequence, X, using a simplified linear Gaussian state-space model in which there is no switching and each signal depends on all other signals:

$$\begin{aligned} X_t^i &= A^i X_{t-1} + n_t^i, \quad & n_t^i &\sim N(0, Q^i) \\ Y_t^i &= X_t^i + v_t^i, & v_t^i &\sim N(0, R). \end{aligned} \tag{4.19}$$

The values of the parameters of this model are sampled from the prior on a single dependence model, assuming that the interaction graph is the full graph. The initial value of the latent time-series state sequence, X, is then generated as a sample from this model. Finally, given initial values of the switching sequence and the state sequence, the initial values of the K dependence models, parameters of the Markov model on the switching sequence, and parameters of the observation model are sampled conditioned on them, as in the full Gibbs sampling procedure.

Gibbs Sampling Schedule

We find that in the examples we explore, it takes a few dozen iterations (e.g., 50-100) for a sampler to converge (to at least a local optimum) and that skipping every few dozen iterations (e.g., 50) to extract a sample results in uncorrelated samples (again, at least conditioned on being in a neighborhood of a local optimum). We typically perform

several restarts to see if there is a significant variation in the results due to different initialization. If that is a case, then we perform multiple restarts (again, typically few dozen) and extract a small number of samples from each of them (typically only one). We also find that even as few as several dozen (e.g., 50) extracted samples can describe the posterior well using the procedure for evaluating the posterior over structures given in the next section.

■ 4.1.3 Evaluating the Posterior

The output of the Gibbs sampling inference algorithm for the SSIM (Algorithm 3.1) is a set of S samples from the joint posteriori over latent variables and model parameters:

$$(\hat{X}^s, \hat{Z}^s, \hat{\pi}^s, \hat{\tilde{E}}^s, \hat{\tilde{\theta}}^s, \hat{\xi}^s), \quad s = 1, \ldots, S, \tag{4.20}$$

where, in sample s, $\hat{X}^s$ is the state sequence, $\hat{Z}^s$ is the switching state sequence, $\hat{\pi}^s$ are the parameters of the Markov model on the switching sequence, $(\hat{\tilde{E}}^s, \hat{\tilde{\theta}}^s) = \{(\hat{\tilde{E}}^s_k, \hat{\tilde{\theta}}^s_k)\}_{k=1}^K$ is a collection of the K dependence models, and $\hat{\xi}^s$ are the parameters of the observation model. Recall that $E_t = \tilde{E}_{Z_t}$ is the interaction structure at time point t in the SSIM. The posterior probability of this structure can be approximated as

$$\begin{aligned} P(E_t = E) = \mathrm{E}[\mathbb{I}(E_t = E)] &\approx \frac{1}{S}\sum_{s=1}^{S} \mathbb{I}(\hat{\tilde{E}}^s_{\hat{Z}^s_t} = E) \\ &= \frac{|\{s : \hat{\tilde{E}}^s_{\hat{Z}^s_t} = E\}|}{S}, \end{aligned} \tag{4.21}$$

where $\mathbb{I}()$ is the indicator function. Note that $\hat{\tilde{E}}^s_{\hat{Z}^s_t}$ is the dependence model indexed by the switching variable at time t in sample s. Also note that the final expression in Equation 4.21 is the fraction among samples of the structures valid at time point t that are equal to E. Furthermore, the posterior probability of any structural event at any time point t, given by an indicator function $f(E_t)$, can be approximated as

$$P(f(E_t) = 1) = \mathrm{E}[f(E_t)] \approx \frac{1}{S}\sum_{s=1}^{S} f(\hat{\tilde{E}}^s_{\hat{Z}^s_t}). \tag{4.22}$$

An example of a structural event indicator is a function $f(E) = \mathbb{I}(1 \rightarrow 2 \in E)$, which indicates whether an edge $1 \rightarrow 2$ exists in the interaction structure E. Then, the probability that signal 1 influences signal 2 at time point t, using Equation 4.22, can

also be written as

$$\begin{aligned} P(1 \to 2 \in E_t) &= \mathrm{E}[\mathbb{I}(1 \to 2 \in E_t)] \\ &\approx \frac{1}{S}\sum_{s=1}^{S} \mathbb{I}(1 \to 2 \in \hat{\tilde{E}}^s_{\hat{Z}^s_t}) \\ &= \frac{|\{s : 1 \to 2 \in \hat{\tilde{E}}^s_{\hat{Z}^s_t}\}|}{S}, \end{aligned} \tag{4.23}$$

which is the fraction among samples of the structures valid at time point t that contain edge $1 \to 2$.

A problem with the above procedure for computing structure posterior probabilities is that the number of possible structures can be very large, even under the modular and bounded in-degree prior assumptions, and many of the structures may not be represented at all in the posterior samples. Similarly, for any low-probability structural event, a very large number of samples is required to estimate that probability reliably. The same holds for conditional events that may have high probability but are conditioned on a low probability event. For example, if one is interested in answering a hypothetical question "What would be the probability that signal 2 influences signal 3 assuming that signal 1 influences signal 3?", but the probability that 1 influences 3 is low, a large number of samples is needed to collect enough samples in which 1 indeed influences 3 in order to estimate the conditional probability.

To alleviate this problem, we estimate the posterior distribution over interaction structure at time point t in the following way:

$$\begin{aligned} P(E_t = E) &= \sum_Z \int_X P(E_t = E, Z, X)\, dX \\ &= \sum_Z \int_X P(E_t = E \mid Z, X) P(Z, X)\, dX \\ &\approx \frac{1}{S}\sum_{s=1}^{S} P(E_t = E \mid \hat{Z}^s, \hat{X}^s) \\ &\approx \frac{1}{S}\sum_{s=1}^{S} P(\tilde{E}_{\hat{Z}^s_t} = E \mid \hat{Z}^s, \hat{X}^s), \end{aligned} \tag{4.24}$$

where $P(\tilde{E}_{\hat{Z}^s_t} = E \mid \hat{Z}^s, \hat{X}^s)$ is the probability distribution over structures of the dependence model indexed by $\hat{Z}^s_t$, conditioned on the state and switching sequences from sample s. Note that computing $P(\tilde{E}_{\hat{Z}^s_t} = E \mid \hat{Z}^s, \hat{X}^s)$ is equivalent to the problem of computing the posterior distribution over a homogenous structure from perfect data, since both the switching pattern and the latent time-series are assumed to be known. Recall that this computation can be done efficiently if a modular bounded in-degree prior on structure and a conjugate prior on parameters of the dependence model are

used, which is the case in this thesis. In other words, samples from the joint posterior distribution in the SSIM are generated first, sampled structures are discarded, and the posterior distribution over structures of each dependence model is then evaluated conditioned on other variables, for each sample. Thus obtained distributions are then averaged for each time point to compute the posterior distribution over structure at that time point. Note that the dependence models do not depend on parameters π and ξ when conditioned on the latent state sequence X and the switching state sequence Z, and thus $\hat{\pi}^s$ and $\hat{\xi}^s$ are irrelevant for computing the posterior over structures as in Equation 4.24. Hyperparameters are omitted from equations for brevity.

Siracusa and Fisher [49, 50] evaluate the posterior distribution over switching interaction structure in the STIM model in the following way. After posterior samples from the joint distribution are generated, a single representative switching sequence, $\hat{Z}$, is determined as the one with the smallest Hamming distance from all samples of switching sequences, $\{\hat{Z}^s\}_{s=1}^S$. Then, the posterior distribution over structure is computed for each switching state exactly, as in the homogenous model (recall that the time-series, X, are assumed known in the STIM). Therefore, their method also resolves the problem of sparse samples of structure. However, the advantage of our method is that the uncertainty in the switching sequence is accounted for when computing the posterior over the interaction structure over time. Furthermore, by marginalizing over the switching sequence, the posterior distribution over structure can be different at every possible time point, whereas in the method of Siracusa and Fisher there are at most K different structures.

Finally, we evaluate the uncertainty in the switching pattern by estimating the probability that any two time points, t_1 and t_2, are in the same switching state:

$$P\left(Z_{t_1} = Z_{t_2}\right) \approx \frac{1}{S} \sum_{s=1}^{S} \mathbb{I}(\hat{Z}_{t_1}^s = \hat{Z}_{t_2}^s)\,. \tag{4.25}$$

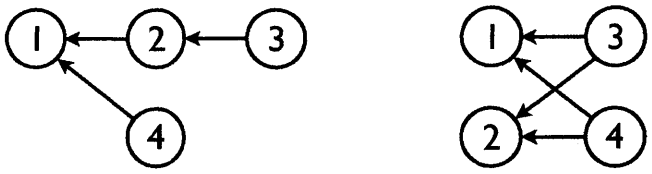
■ 4.2 Synthetic Data Experiments

We present several experiments with synthetic data that test different aspects of the interaction structure learning problem.

■ 4.2.1 Structure Inference vs. Pairwise Test

Learning interaction graphs under the modular prior assumption (Section 2.6.4) in general requires testing each possible parent set of each node. If parents of a node are tested individually, the most likely parents may not necessarily be the correct ones. To demonstrate that, we generate two examples from the LG-SSIM model. In both examples, there is no switching and observations are assumed perfect. There are 4 univariate signals in both examples, and the first-order AR model is assumed.

The interaction structure for the first example is shown in Figure 4.1a, and the



(a) (b)

Figure 4.1: The interaction structure in the two examples that demonstrate the necessity to consider parent sets rather than parent candidates individually.

linear Gaussian models for each node are:

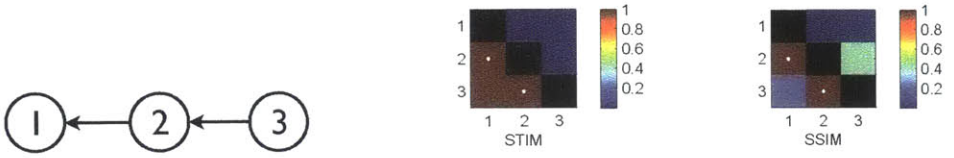
$$\begin{aligned} X_t^1 &= 0.2X_{t-1}^1 + 0.6X_{t-1}^2 + 0.1X_{t-1}^4 + n_t^1, \\ X_t^2 &= 0.2X_{t-1}^2 + 0.7X_{t-1}^3 + n_t^2, \\ X_t^3 &= 0.9X_{t-1}^3 + n_t^3, \\ X_t^4 &= 0.9X_{t-1}^4 + n_t^4, \end{aligned} \tag{4.26}$$

where n_t^1, n_t^2, n_t^3 and n_t^4 are I.I.D. samples from $\mathcal{N}(0, 0.1)$. A data sequence of length $T = 1000$ is sampled from this model.

When the inference is performed on these data without any restrictions on possible parents (except for self-dependencies, which are always assumed), the correct structure is recovered (posterior probabilities of true and false edges are approximately equal to 1 and 0, respectively). However, by looking at the effect of each signal separately, in addition to self-dependency, possible parents of signal 1 are sorted as 2, 3, 4, in the order of decreasing posterior likelihood. Therefore, the posterior likelihood of the false edge $1 \leftarrow 3$ is higher than the likelihood of true edge $1 \leftarrow 4$. This result stems from the fact that the true dependency of signal 1 on signal 4 is relatively weak (with coefficient 0.1), while its indirect dependency on signal 3 is stronger. Note that this is essentially a test for Granger causality [23].[2] The test is performed in the LG-SSIM model simply by bounding the number of parents to 2 (i.e., 1 in addition to the assumed self-dependency). Finally, if signal 2 is excluded from the analysis, but there is no restriction on the number of other parents, the posterior probability of edges $1 \leftarrow 3$ and $1 \leftarrow 4$ are 1 and 0.7, respectively. This shows that, in the absence of signal 2, signal 3 "takes over" its role in explaining signal 1, together with signal 4. Since the relationship between signals 1 and 3 is noisier than between signals 1 and 2, the probability of this explanation is lower (probability of edge $1 \leftarrow 3$ equal to 1 means that the probability of 3 alone being a parent is $1 - 0.7 = 0.3$, while the probability of a parent set $\{3, 4\}$ is 0.7).

In the second example, a sequence of length $T = 1000$ is sampled from the model with the interaction structure shown in Figure 4.1b and the following linear Gaussian

[2]Except that the parameters are marginalized out instead of looking at the maximum-likelihood parameters.



(a) (b) (c)

Figure 4.2: An example that demonstrates the advantage of modeling observation noise. (a) True interaction structure. (b) Posterior probability of edges obtained by inference in the STIM model (which does not model observation noise). (b) Posterior probability of edges obtained by inference in the SSIM model (which models observation noise). The value at row i and column j is the probability of edge $i \to j$. Self-edges are blacked out, while the correct edges are marked with a white dot. Note that the STIM assigns probability 1 to a false edge $1 \leftarrow 3$. Even though signal 1 depends only indirectly on signal 3 in the generative model, signal 3 helps explain signal 1 since the observations of signal 2 are noisy. On the other hand, if the SSIM is used for inference, the posterior probability of edge $1 \leftarrow 3$ is significantly reduced. Note also that the probability of edge $3 \leftarrow 2$ has increased, which means that the additional flexibility of the model may allow for different explanation of the data in the latent space.

models:

$$\begin{aligned} X_t^1 &= 0.1X_{t-1}^1 + 0.4X_{t-1}^3 + 0.4X_{t-1}^4 + n_t^1\,, \\ X_t^2 &= 0.1X_{t-1}^2 + 0.4X_{t-1}^3 + 0.4X_{t-1}^4 + n_t^2\,, \\ X_t^3 &= 0.9X_{t-1}^3 + n_t^3\,, \\ X_t^4 &= 0.9X_{t-1}^4 + n_t^4\,, \end{aligned} \tag{4.27}$$

where n_t^1, n_t^2, n_t^3 and n_t^4 are I.I.D. samples from $\mathcal{N}(0, 0.1)$. Again, when all parent sets are considered, inference yields the posterior probability of the correct structure approximately equal to 1. However, when parents are considered individually, the most likely parent of node 2 is node 1, with probability 0.94. This can be explained by the fact that both signal 1 and signal 2 depend on signals 3 and 4 in the same way, and can thus be similar to each other. In this case, it happens that signal 1 helps predict signal 2 better than either of signals 3 and 4 individually.

■ 4.2.2 Observation Noise vs. No Observation Noise

We demonstrate the advantage of the SSIM model over the STIM model of Siracusa and Fisher [49, 50] described in Section 2.7, which does not account for observation noise. We generate an example from the LG-SSIM model in which there are 3 univariate signals, there is no switching, and the first-order AR model is assumed. A sequence of length $T = 1000$ of 3 signals is sampled from the LG-SSIM model with the interaction

structure shown in Figure 4.2a and the following first-order linear Gaussian models:

$$\begin{aligned} X_t^1 &= 0.2X_{t-1}^1 + 0.7X_{t-1}^2 + v_t^1 , \\ X_t^2 &= 0.2X_{t-1}^2 + 0.7X_{t-1}^3 + v_t^2 , \\ X_t^3 &= 0.9X_{t-1}^3 + v_t^3 , \end{aligned} \tag{4.28}$$

where v_t^1, v_t^2 and v_t^3 are I.I.D. samples from $\mathcal{N}(0, 0.1)$. There is no switching. Signals 1 and 3 are observed directly, while signal 2 is observed via a noisy process:

$$Y_t^2 = X_t^2 + w_t^2 , \qquad w_t^2 \sim \mathcal{N}(0, 0.1) . \tag{4.29}$$

When the linear Gaussian STIM model, which assumes perfect observations, is used for inference, the posterior probability of edges is shown in Figure 4.2b. Note that the probability of a false edge $1 \leftarrow 3$ is 1. Even though signal 1 depends directly on signal 2 and only indirectly on signal 3 in the generative model, the observed signal 3 helps explain signal 1 since the observations of signal 2 are noisy. On the other hand, if the LG-SSIM model, in which observation noise is allowed, is used for inference, the posterior probability of edges is shown in Figure 4.2c. Clearly, the posterior probability of edge $1 \leftarrow 3$ is significantly reduced. Note that the probability of edge $3 \leftarrow 2$ has increased, which means that the additional flexibility of the model may allow for different explanation of the data. In this case, some of the posterior probability mass is centered on the explanation in which signal 3 depends on signal 2. Still, the most certain edges are the correct ones. In addition, the expected value of the latent signal 2 in the posterior distribution is closer to its true value than the observed signal is in terms of L_1 and L_2 norm.

■ 4.3 Joystick Interaction Game

Most available temporal data is not annotated for interactions. Furthermore, obtaining ground truth interactions is difficult and, in most cases, subjective. While that amplifies the importance of developing algorithms that aid in uncovering such interactions, it also makes the testing of these algorithms difficult. Consequently, we created a simple experiment, from so-called "joystick" data, where the structure is known (although the parameterization is not). In the experiment, five players control a joystick to move an object on the screen in order to accomplish a task. There are three different assignments of tasks shown in the top of Figure 4.3. Assignments switch over time over the duration of 4.5 minutes, as shown in the bottom of the figure. To remove bias, a player only sees the objects on which it depends. Positional (2D) data is recorded every 1/10sec., so there is a total of 2701 time points, including the initial one. This data is realistic since it is human-generated and not synthesised from the model. In addition, it contains interaction annotations by design and is useful for validating the model.

We find that the best results are obtained when the data is downsampled 3 times (total of 901 time points) and AR order is 5, which we use in all experiments. This order

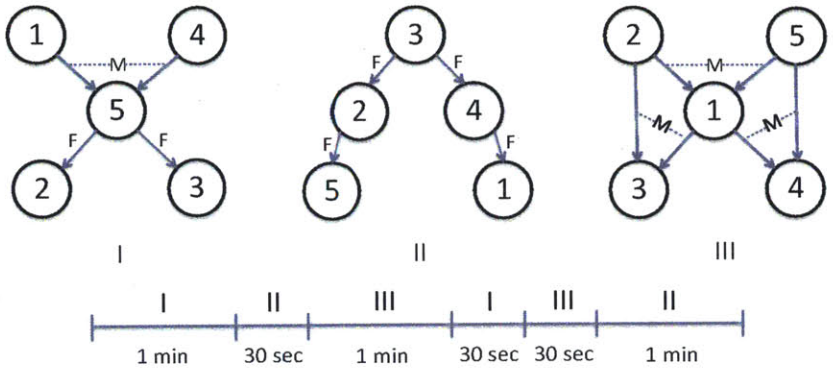


Figure 4.3: (top) Three assignments of tasks. Individual tasks can be: F – "follow", M – "stay in the middle between", and "move arbitrarily" (otherwise). (bottom) Order and duration of assignments.

corresponds to a lag of 1.5 seconds. A 3 times higher AR order would be required with the original data in order to capture the dependencies of the same length. However, the original data does not provide much additional information due to high correlation of samples at neighboring time points.

In all of the experiments, self-dependencies are assumed and are included in the count of parents. Results with $K = 3$, $b = 10$, and maximum number of parents set to 3 and 5, respectively, are shown in Figure 4.4. The top row presents the switching-state pairwise probability matrix, whose entry (i, j) is the posterior probability that time points i and j are assigned the same switching state. There is an obvious switching pattern that coincides with the setup of the experiment. The bottom row shows the posterior probabilities of edges at 0.5, 1.25 and 2 min, which correspond to the three different assignments. The value in i^{th} row and j^{th} column is the probability of edge $i \to j$. Self-edges are "blacked out", while the assignment ("correct") edges are marked with a white dot. The algorithm assigns high posterior probability to all correct edges. In addition, a few spurious edges are assigned medium to high probability. We note that these are typically edges between players that follow a common other player, possibly via intermediate players. For example., 2 and 3 both follow 5 in the first assignment, while 4 and 5 (via 2) both follow 3 in the second assignment. We also note that the results are better when fewer parents are allowed, since the number of possible incorrect choices of parents is reduced.

We set maximum number of parents to 3 in the rest of the experiments. Interestingly, when only two switching states are allowed, the switching pattern still indicates the presence of three states, as shown in Figure 4.5. Namely, states 1 and 2 are combined into a single state in some samples, while states 2 and 3 are combined in other samples. On the other hand, when $K = 5$ states are allowed, only 3 of them are actually used, yielding similar results as with $K = 3$.

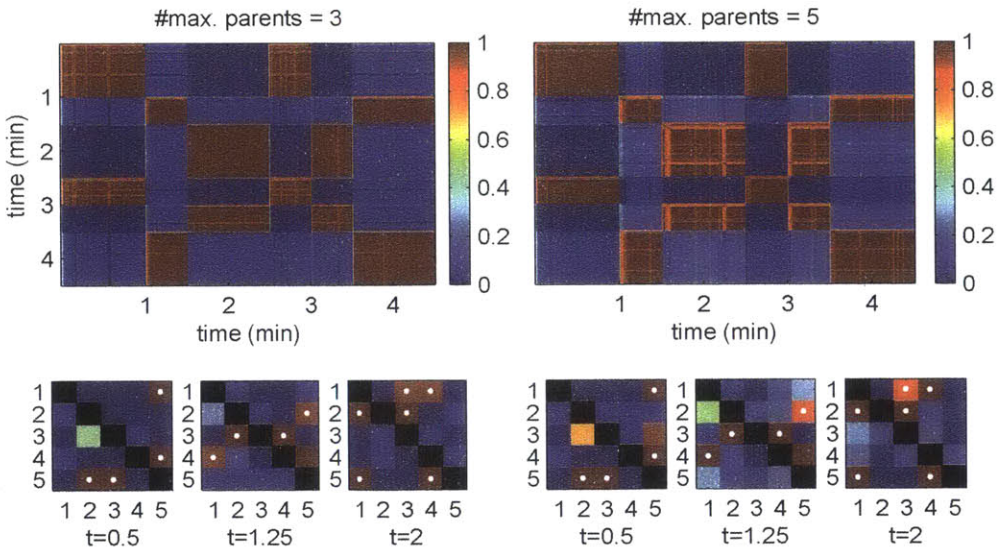


Figure 4.4: Interaction analysis on Joystick data when the maximum number of parents is 3 (left) and 5 (right). Top row are the switching-state pairwise probability matrices. Value at a position (t_1, t_2) is the probability that time points t_1 and t_2 are assigned the same switching state, i.e., $P(Z_{t_1} = Z_{t_2})$. Note that in both cases there is an obvious switching pattern that coincides with the setup of the experiment. A red block on the diagonal shows high probability that the corresponding time segment is homogenous in terms of interaction (i.e., corresponds to a single switching state). A red off-diagonal block shows that time segments corresponding to its projections onto x and y axes have the same interaction (are in the same switching state). Bottom row are edge posterior matrices at times 0.5, 1.25 and 2 min, which correspond to the three different assignments. The value at row i and column j is the probability of edge $i \to j$. Self-edges are blacked out, while the correct edges are marked with a white dot. Note that the SSIM assigns high probability to all correct edges and to a few spurious edges. Those errors commonly occur when two players have very similar behavior (e.g., players 2 and 3 both follow player 5 in the first assignment). Note also that there results are slightly worse when the maximum number of parents is 5, which is higher than needed.

Finally, we test our algorithm in the scenarios of higher uncertainty. In the first experiment, we add Gaussian noise of a fixed variance to all observations. Selection of variance 10^{-5} does not change the results.[3] The results with variance 10^{-4} show higher uncertainty in some of the edges (Figure 4.6, left). Also, from the switching pattern we see that states 2 and 3 are not distinguished from each other in some of the samples. When noise variance is further increased to 10^{-3}, none of the three states is recognized. In the second experiment, we treat a subset of the data as missing. When

[3]The maximum distance an object can travel between two time points is 0.075.

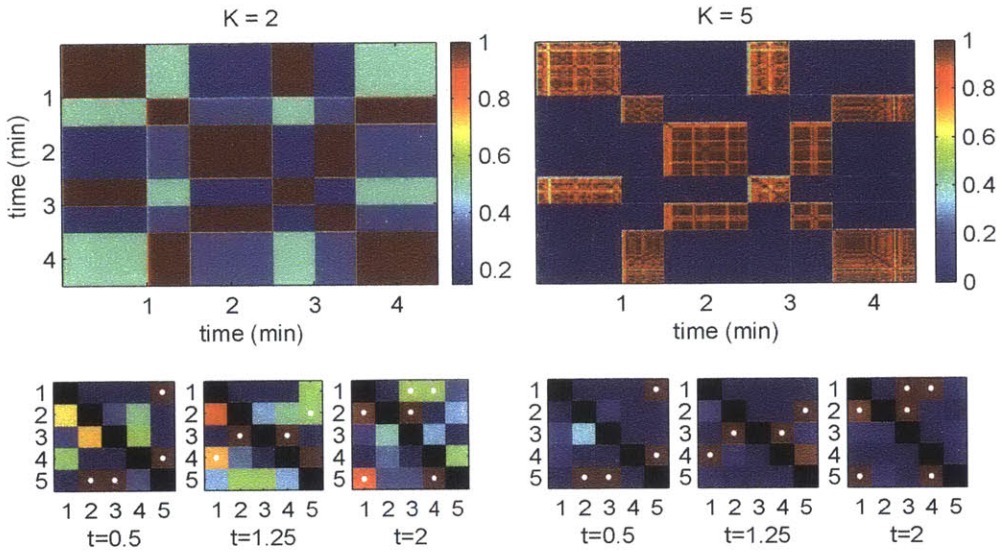


Figure 4.5: Results on Joystick data when the number of switching state K is 2 (left) and 5 (right). Top row are switching similarity matrices. Bottom row are edge posteriors at times 0.5, 1.25 and 2 min. Note that even when K is lower than the actual number of switching states ($K = 2$), the switching similarity matrix indicates the presence of 3 states, and there are also three distinct interaction structures. The first result highlights the advantage of looking at the entire posterior distribution rather than at a MAP assignment. The second result is due to marginalization of the switching state sequence. Note also that when K is higher than the actual number of switching states ($K = 5$), the results are similar to those obtained with the correct number of states (Figure 4.4, left), which indicates that the additional states allowed are not assigned any new behavior that consistently appears in a large number of samples.

every 2^{nd} value is observed, the results do not change. The results when every 3^{rd} value is observed (Figure 4.6, right) show higher uncertainty of some edges.

■ 4.3.1 Comparison to other approaches

We illustrate the advantage of the SSIM model, which accounts for the observation noise, over the previous model of Siracusa and Fisher [50] (STIM), which assumes perfect observations. A subset of the joystick data that correspond to the last 1-minute segment is taken and Gaussian noise with variance 10^{-3} (high noise) is added to all observations. Note that there is no switching during this segment and the correct interaction structure is that of the second assignment in Figure 4.3. Posterior edge probabilities obtained by inference using the SSIM and STIM models with a single switching state are shown in Figure 4.7. The STIM model assigns high probability to only one correct edge and does not infer other edges due to high observation noise. On

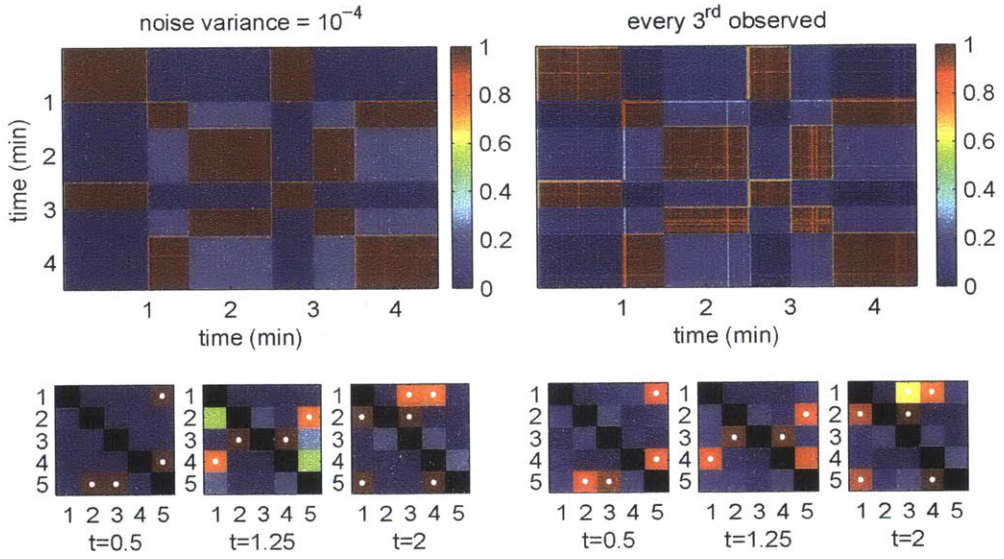


Figure 4.6: Results on Joystick data when observation noise variance is 10^{-4} (left) and when every 3^{rd} value is observed (right). Top row are switching similarity matrices. Bottom row are edge posteriors at times 0.5, 1.25 and 2 min. Note that these results are qualitatively similar to those obtained from perfect data (Figure 4.4, left), even though relatively high noise is added to observation in one case and a large fraction (2/3) of observations are dropped in the second case. The uncertainty in the observation sequence is reflected in the posterior as a (slightly) higher uncertainty in the interaction structures and the switching pattern.

the other hand, the SSIM model assigns high probability to 3 out of 4 correct edges, which is an evidence that it can infer interactions among the latent time-series that are not detectable from the observed time-series directly.

Note that while the SSIM model assigns significant posterior probability to incorrect edges $3 \rightarrow 5$ and $4 \rightarrow 5$, there is some uncertainty in these edges. This is to compare with the MAP estimate of the structure, also shown in Figure 4.7, which simply presents a single most likely parent set of node 5 (as well as for other nodes) and does not account for the uncertainty in the estimated structure.

■ 4.4 Climate Indices Interaction Analysis

Here, we apply the LG-SSIM model to real-world climate data. In doing so, we wish to emphasize that one should be careful in drawing scientific conclusions from these results. In particular, the interactions amongst these data sets are likely not linear (as assumed by the LG-SSIM) and consequently, inferred structures may not necessarily be indicative of explicit causality. Nevertheless, the analysis may yield interesting details.

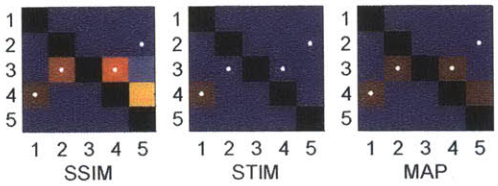


Figure 4.7: Results of structure inference on a segment of Joystick data that corresponds to the second assignment (no switching), and to which high noise is added (variance of 10^{-3}), obtained via: full inference in the SSIM model (left), full inference in the STIM model of Siracusa and Fisher [50] that does not account for the observation noise (middle), and MAP estimate in the SSIM model (right). Note that the SSIM assigns high probability to 3 out of 4 correct edges, while the STIM assigns high probability to only one of them. Also note that the SSIM assigns a reduced probability (higher uncertainty) to the incorrect edge in the MAP structure (edge $4 \rightarrow 5$).

Following Jiang et al. [31], we use data on a subset of 16 climate indices from the repository maintained by the Earth System Research Laboratory of the National Oceanic and Atmospheric Administration (NOAA) [40], which are described in Table 4.1. These indices are compiled monthly and span various characteristics of the climate system. For the purpose of comparison, we use the data from 1951 to 2007, as in Jiang et al., and apply linear and quadratic detrending. Note that a small fraction of the data in this span is missing, which our model addresses naturally.

#	abbrev.	description
1	AMM	Atlantic Meridional Mode SST
2	AO	Arctic Oscillation
3	EP/NP	East Pacific/North Pacific Oscillation
4	GMT	Global Mean Lan/Ocean Temperature
5	Nino3	Eastern Tropical Pacific SST
6	Nino4	Central Tropical Pacific SST
7	Nino12	Extreme Eastern Tropical Pacific SST
8	Nino34	East Central Tropical Pacific SST
9	NOI	Northern Oscillation Index
10	ONI	Oceanic Nino Index
11	PDO	Pacific Decadal Oscillation
12	PNA	Pacific North American Index
13	SOI	Southern Oscillation Index
14	Solar	Solar Flux (10.7cm)
15	SWM	South Western USA Monsoon
16	WP	Western Pacific Index

Table 4.1: Description of climate indices.

We run inference using the SSIM latent-AR model with two switching states. We

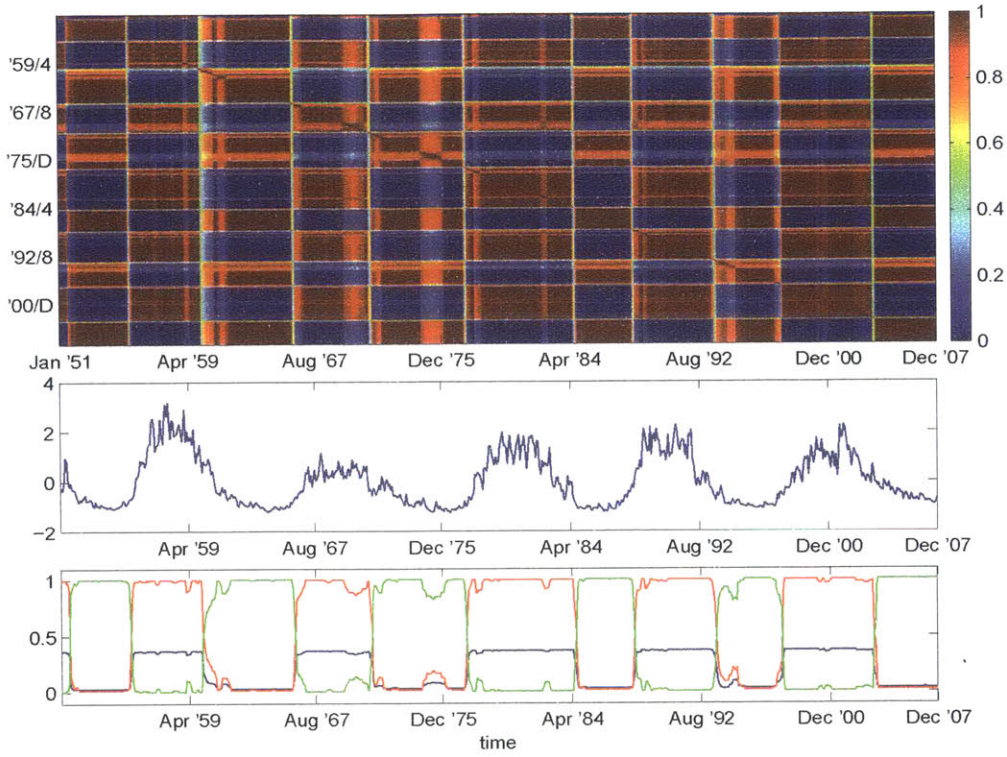


Figure 4.8: Analysis of the climate data using SSIM model. Top row is the switching-state pairwise probability matrix. Middle row is the Solar flux time series. Bottom row are the posterior probabilities of edges: Nino12 → GMT (blue), Nino12 → Nino4 (red), Nino12 → Nino34 (green). Note that the switching pattern exhibits a cyclic behavior that coincides with the cycles of Solar flux. The two states correspond to the low and high activity of Solar flux.

bound the number of parents per node to 3 and require a minimum of 1 parent with enforcing self-edges. The top row of Figure 4.8 shows the switching-state pairwise probability matrix. Unlike Jiang et al., whose results suggest a single switch point in 1978, this result suggests that there is a cyclic behavior. Figure 4.10 shows two matrices of posterior probabilities of edges that correspond to June 1963 (left) and August 1992 (right), which belong to the opposite phases of the cycle. We observe that Nino indices and ONI index are the most influential overall, confirming that they are important predictors of climate [60]. Interestingly, the only significant dependence of ONI index is on Southern Oscillation Index.

Note that there are a few differences between the two posteriors. For example, as shown in the bottom row of Figure 4.8, influence of Nino12 index onto GMT, Nino4 and Nino34 indices fluctuates dramatically. As noted above, these may not necessarily be changes in explicit causality. Still, they represent the best explanations of the structural

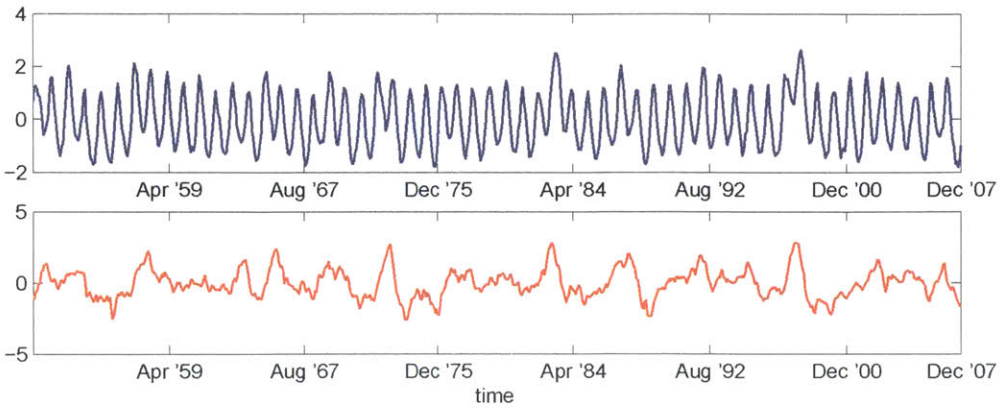


Figure 4.9: Nino12 (top) and ONI (bottom) time series.

dependencies in the two phases under the LG-SSIM model. In addition, the ambiguity in the switching pattern between regimes may suggest that there exist transition periods of several months to several years, rather than a sharp change. This may explain the differences in the switchpoints reported in the literature [29, 31], emphasizing the advantage of Bayesian reasoning over point estimation.

Unlike Jiang et al. [31], in which Solar flux is the most influential index, the results obtained here show no direct dependency on Solar flux, but suggest its indirect influence via the switching state. Namely, we observe that the switching sequence largely corresponds to the change of variance of Solar flux and that it is likely that a more complex, nonlinear model describes it's exact relationship to the remaining indices. Interestingly, the Nino12 index does not appear to correlate with the switching pattern (Figure 4.9); however, its influence on the three other indices changes according to the behavior of Solar flux. The same holds for other time series (e.g., ONI, also shown in Figure 4.9).

Finally, we note that the exact nature and magnitude of the influence of Solar variability on the climate is still largely unknown [26, 34] and presents an active area of research. It is particularly hard to distinguish the Solar influence from that of greenhouse gases and aerosols in the industrial era, to which the data used here belongs. Therefore, it is not surprising that we do not discover direct short-term linear dependency of climate indices on Solar flux, suggesting that using a nonlinear model and data over a longer period of time or at a different time scale may be more adequate for that particular task.

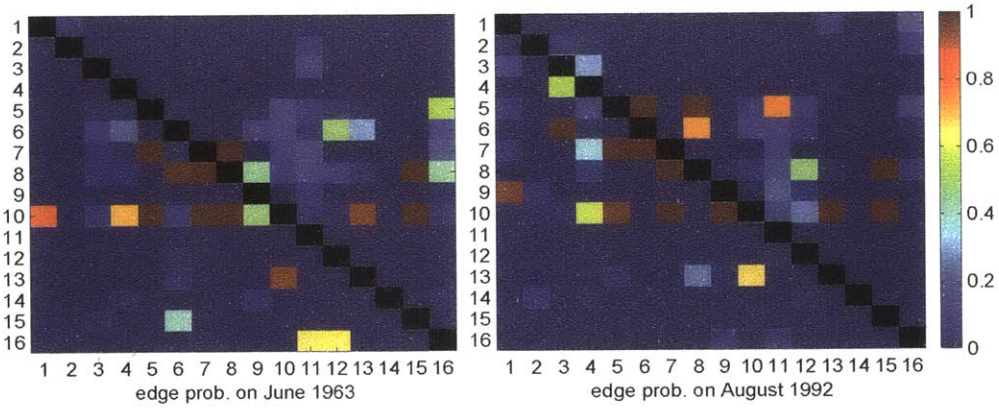


Figure 4.10: Posterior edge probabilities on June 1963 (left) and August 1992 (right), which belong to the opposite phases of the cycle. Note that Nino indices (5-8) and ONI index (10) are the most influential overall, confirming that they are important predictors of climate. Interestingly, the only significant dependence of ONI index is on Southern Oscillation Index (13).

Chapter 5

Structural Health Monitoring with SSIM

STRUCTURAL inspection has been necessary to ensure the integrity of infrastructure for almost as long as structures have existed, ranging from informal subjective methods such as visual or hammer testing, to quantitative modern methods including ultrasound, x-ray, and radar non-destructive testing techniques. These testing methods are relatively intensive as they depend on the experience of the inspector and the time to inspect suspected damaged locations in the structure. Inspections are typically carried out periodically, however if additional sensors could be added to the structure such that some indication of where potential locations of damage might be such that they can be closely inspected, it would be useful for reducing the time and effort necessary for structural inspection.

Structural health monitoring (SHM) involves instrumenting a structure with sensors and deriving some information from the data they collect in order to determine if the structure has changed [6]. This change in the structure could then be attributed to some sort of damage that would be more closely investigated. In general, data is processed into features that may indicate these changes in the structure and in some cases statistical or probabilistic discrimination of these features are used to separate data collected from intact and changed structures [51]. Statistical methods are essential for being able to discriminate feature changes as a result of structural changes from measurement or environmental variability.

Bayesian inference can be used in a couple of different ways in SHM including model updating of structural parameters [4], monitoring by inferring structural parameters over time [56], and determining the optimal placement of sensors [15]. Bayesian inference can be used in either a model-based situation where a structural model is either formulated or updated as a basis for damage detection, a data-based situation where there is no prior information on the structural model and only the sensor data is used, or a mixture of the two situations.

We apply the SSIM framwork to time-series data obtained from accelerometers located at multiple positions on a building. By accounting for interactions between sensor signals collected from the system in different locations, the hope is to infer a

representation of the structural connections between locations in the structure or the underlying physics without having any knowledge of the actual structural configuration or dynamics. Assuming that the model learned from a set of data is exclusive to the corresponding physical structural configuration and condition, a change in the model parameters could be indicative of a change in the measured physical structure which might be caused by damage. In order to see if these assumptions might hold true, we test the methodology on data from model structures in various intact and damaged conditions, as well as on data from a real building under ambient and non-ambient conditions, such as fireworks and earthquake. These data consist of short sequences of measurements, and it can be assumed that changes do not occur within a single sequence. The problem of damage detection can then be cast as a problem of time-series classification. If prior data from possible damage scenarios is available, then this problem is a standard multi-class classification problem. However, in most real scenarios, only data from an intact structure is available a priori. Then, the problem of damage detection can be seen as a single-class classification problem.

We introduce the SSIM model for classification of time-series in Section 5.1 and its single-class classification variant in Section 5.2. We describe the data and experimental results on two laboratory model structures in Section 5.3 and MIT Green building in Section 5.4. We perform interaction analysis on both datasets and show that inferred edges correlate with an actual physical structure. On the laboratory data, we demonstrate that the SSIM classification model can classify time-series obtained under intact and different damage scenarios with high accuracy, in both standard and single-class classification settings. Finally, on the MIT Green building data, we demonstrate that the SSIM single-class classification model can distinguish time-series obtained under conditions that differ form ambient conditions (from those obtained under ambient conditions) and that it also predicts the "strength of deviation".

■ 5.1 Classification with SSIM

The SSIM model can simply be extended to multiple sequences, as shown in Figure 5.1. Here, L denotes the number of sequences. Each observation sequence $\mathcal{Y}_l = (Y_{l0}, Y_{l1}, \ldots, Y_{lT_l})$ has an associated state sequence $\mathcal{X}_l = (X_{l0}, X_{l1}, \ldots, X_{lT_l})$ and switching sequence $\mathcal{Z}_l = (Z_{l1}, Z_{l2}, \ldots, Z_{lT_l})$, where l is a sequence index and T_l denotes the length of sequence l. The inference is still performed as in Algorithm 3.1, except that steps 1 and 2 are repeated for each sequence separately, while the data needed in steps 3, 4, and 5 (i.e., values of X, Y and Z) is pulled from all sequences. We will use $\mathcal{Y} = \{\mathcal{Y}_l\}_{l=1}^{L}$, $\mathcal{X} = \{\mathcal{X}_l\}_{l=1}^{L}$ and $\mathcal{Z} = \{\mathcal{Z}_l\}_{l=1}^{L}$ to denote collections of observation, latent state and switching sequences, respectively.

In some scenarios, changes in behavior (dependence model) are only expected across different sequences, but not within each sequence. For example, this is the case in a damage detection setup that we exploit, in which short sequences of measurements (e.g., $\sim$ 1min) are recorded far apart from each other (e.g., $\sim$ 1hour). Sequences are

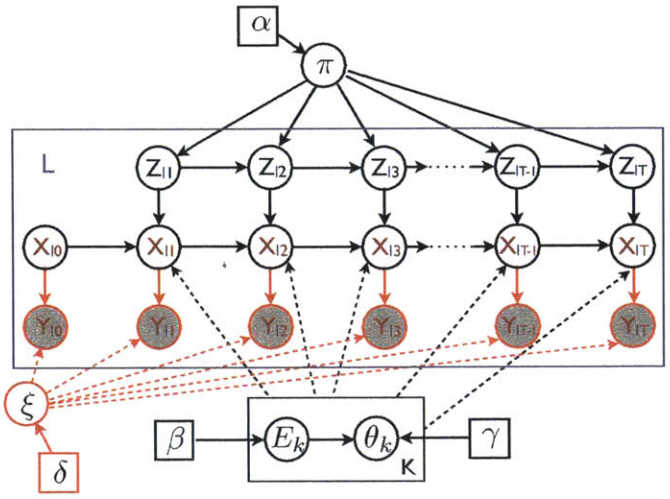


Figure 5.1: SSIM model with multiple sequences.

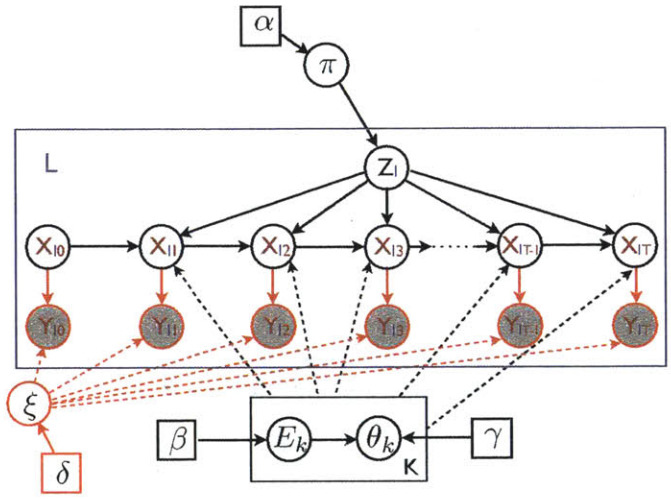


Figure 5.2: SSIM model with multiple homogenous sequences.

short enough such that changes within them are unlikely. If switching does not occur within sequences, then each sequence can be assigned a single switching state variable, Z_l. We refer to this model as SSIM with multiple homogenous sequences, which is shown in Figure 5.2. Since there are no transitions between switching states, this model does not require transition probabilities and parameters of their corresponding Dirichlet priors. Only initial probabilities are needed, and thus $\pi = (\pi_1, \ldots, \pi_K)$ and $\alpha = (\alpha_1, \ldots, \alpha_K)$. In this context, we will refer to sequence switching states as sequence labels. Inference over switching states (labels) in this model is essentially inference over clusters of sequences according to their dynamics (i.e., dependence model).

Classification of sequences can be reduced to the inference in SSIM with multiple homogenous sequences by performing joint inference over training sequences and a test sequence while fixing the labels of training sequences. The probability of any value of the test sequence label is then the frequency of that value in the posterior samples. However, these probabilities can be computed more directly in the following way.

We assume that in a classification problem there are K classes, and, for each class $k \in \{1, 2, \ldots, K\}$, a collection of N_k^{tr} training sequences $\mathcal{Y}_k^{tr} = \{\mathcal{Y}_{kj}^{tr}\}_{j=1}^{N_k^{tr}}$ is given, thus implicitly assuming $Z_{kj}^{tr} = k$ for each j. In addition, we will use $\mathcal{Z}_k^{tr} = \{Z_{kj}^{tr}\}_{j=1}^{N_k^{tr}} = \{k\}^{N_k^{tr}}$ to denote a collection of labels associated with training sequences from class k, where $\{k\}^{N_k^{tr}}$ denotes a collection of N_k^{tr} values equal to k. Given a test sequence $\mathcal{Y}^{test}$ and the training data, the goal is to find the probability distribution of the test sequence label, i.e., $P(Z^{test} = k \mid \mathcal{Y}^{test}, \{\mathcal{Y}_{k'}^{tr}, \mathcal{Z}_{k'}^{tr}\}_{k'=1}^{K})$, for each k. This probability can be computed in the following way:[1]

$$
\begin{aligned}
& P(Z^{test} = k \mid \mathcal{Y}^{test}, \{\mathcal{Y}_{k'}^{tr}, \mathcal{Z}_{k'}^{tr}\}_{k'=1}^{K}) \\
& \propto P(Z^{test} = k, \mathcal{Y}^{test} \mid \{\mathcal{Y}_{k'}^{tr}, \mathcal{Z}_{k'}^{tr}\}_{k'=1}^{K}) \\
& = P(Z^{test} = k \mid \{\mathcal{Y}_{k'}^{tr}, \mathcal{Z}_{k'}^{tr}\}_{k'=1}^{K}) \cdot P(\mathcal{Y}^{test} \mid Z^{test} = k, \{\mathcal{Y}_{k'}^{tr}, \mathcal{Z}_{k'}^{tr}\}_{k'=1}^{K}) \\
& = P(Z^{test} = k \mid \{\mathcal{Z}_{k'}^{tr}\}_{k'=1}^{K}) \cdot P(\mathcal{Y}^{test} \mid Z^{test} = k, \mathcal{Y}_k^{tr}, \mathcal{Z}_k^{tr}) .
\end{aligned}
\tag{5.1}
$$

The last equality follows from the fact that the test label is independent of the training sequences given training labels, and that the test sequence, assuming it belongs to class k, only depends on the training data for that class.

The first term in Equation 5.1, $P(Z^{test} = k \mid \{\mathcal{Z}_{k'}^{tr}\}_{k'=1}^{K})$, is the probability of a test sequence belonging to class k before seeing the sequence, given training labels. It can be computed by marginalizing out multinomial parameters π:

$$
\begin{aligned}
P(Z^{test} = k \mid \{\mathcal{Z}_{k'}^{tr}\}_{k'=1}^{K}) \equiv P(Z^{test} = k \mid \{\mathcal{Z}_{k'}^{tr}\}_{k'=1}^{K}; \alpha) & \\
= \int_{\pi} P(Z^{test} = k \mid \pi)\, P(\pi \mid \{\mathcal{Z}_{k'}^{tr}\}_{k'=1}^{K}; \alpha)\, d\pi & \\
= \int_{\pi} \pi_k \cdot \mathcal{D}ir(\pi; \alpha_1 + N_1^{tr}, \alpha_2 + N_2^{tr}, \ldots, \alpha_K + N_K^{tr})\, d\pi & \\
= \frac{\alpha_k + N_k^{tr}}{\sum_{k'=1}^{K} \alpha_{k'} + N_{k'}^{tr}} . &
\end{aligned}
\tag{5.2}
$$

Note that $P(\pi \mid \{\mathcal{Z}_{k'}^{tr}\}_{k'=1}^{K}; \alpha)$ is the posterior distribution of π given training labels, which is again a Dirichlet distribution (with updated parameters) due to conjugacy. The final expression is obtained as the expectation of parameter π_k with respect to that distribution. For convenience, we will write $P(Z^{test} = k \mid \{\mathcal{Z}_{k'}^{tr}\}_{k'=1}^{K}) \equiv P_{tr}(Z^{test} = k)$.

[1] In this section, hyperparameters are omitted for brevity, but will be reinserted as needed.

The second term in Equation 5.1, $P(\mathcal{Y}^{test} \mid Z^{test} = k, \mathcal{Y}_k^{tr}, Z_k^{tr})$, is the marginal likelihood of a test sequence under the class k model, given the training sequences $\mathcal{Y}_k^{tr}$ from that class. It is computed by marginalizing out k^{th} model structure and parameters (model averaging):

$$P(\mathcal{Y}^{test} \mid Z^{test} = k, \mathcal{Y}_k^{tr}, Z_k^{tr}) = \sum_{\tilde{E}_k} \int_{\tilde{\theta}_k} P(\mathcal{Y}^{test} \mid \tilde{E}_k, \tilde{\theta}_k)\, P(\tilde{E}_k, \tilde{\theta}_k \mid \mathcal{Y}_k^{tr})\, d\tilde{\theta}_k \,. \tag{5.3}$$

The term $P(\tilde{E}_k, \tilde{\theta}_k \mid \mathcal{Y}_k^{tr})$ is the posterior distribution of k^{th} model structure and parameters given the training sequences $\mathcal{Y}_k^{tr}$, which then serves as a prior for evaluating the test sequence likelihood. For convenience, we will write $P(\mathcal{Y}^{test} \mid Z^{test} = k, \mathcal{Y}_k^{tr}, Z_k^{tr}) \equiv \mathcal{L}_k(\mathcal{Y}^{test} \mid \mathcal{Y}_k^{tr})$.

Finally, the posterior distribution of the test sequence label, Z^{test}, is obtained by normalizing Equation 5.1:

$$P(Z^{test} = k \mid \mathcal{Y}^{test}, \{\mathcal{Y}_{k'}^{tr}, Z_{k'}^{tr}\}_{k'=1}^{K}) = \frac{P_{tr}(Z^{test} = k)\, \mathcal{L}_k(\mathcal{Y}^{test} \mid \mathcal{Y}_k^{tr})}{\sum_{k'=1}^{K} P_{tr}(Z^{test} = k')\, \mathcal{L}_{k'}(\mathcal{Y}^{test} \mid \mathcal{Y}_{k'}^{tr})} \,. \tag{5.4}$$

The maximum a posteriori (MAP) estimate is obtained as

$$\begin{aligned} \hat{Z}^{test} &= \arg\max_k P(Z^{test} = k \mid \mathcal{Y}^{test}, \{\mathcal{Y}_{k'}^{tr}, Z_{k'}^{tr}\}_{k'=1}^{K}) \\ &= \arg\max_k P_{tr}(Z^{test} = k)\, \mathcal{L}_k(\mathcal{Y}^{test} \mid \mathcal{Y}_k^{tr}) \,. \end{aligned} \tag{5.5}$$

Computing the likelihood in Eq. 5.3 in closed form is intractable in general. The latent training and test state sequences, $\mathcal{X}_k^{tr}$ and $\mathcal{X}^{test}$, need to be marginalized out to compute $P(\tilde{E}_k, \tilde{\theta}_k \mid \mathcal{Y}_k^{tr})$ and $P(\mathcal{Y}^{test} \mid \tilde{E}_k, \tilde{\theta}_k)$, respectively, and simultaneous marginalization of a state sequence and model structure and parameters is analytically intractable. Instead, this likelihood can be computed via simulation:

$$\mathcal{L}_k(\mathcal{Y}^{test} \mid \mathcal{Y}_k^{tr}) \approx \frac{1}{N_s} \sum_{j=1}^{N_s} P(\mathcal{Y}^{test} \mid \hat{E}_j, \hat{\theta}_j) \,, \quad (\hat{E}_j, \hat{\theta}_j) \sim P(\tilde{E}_k, \tilde{\theta}_k \mid \mathcal{Y}_k^{tr}) \,. \tag{5.6}$$

N_s instances of dependence models, $(\hat{E}_j, \hat{\theta}_j)$, are sampled from the posterior distribution of the k^{th} model given training sequences. The test sequence likelihood is evaluated against each of the sampled models, and then averaged out. On the other hand, in an approximate model which assumes no observation noise (i.e., $\mathcal{X}_i \equiv \mathcal{Y}_i$), the likelihood in Eq. 5.3 can be computed in closed form by updating the conjugate prior on dependence structure and parameters with the training data and then evaluating the likelihood of the test data against thus obtained posterior.

■ 5.2 Single-Class Classification with SSIM

In a typical real structural health monitoring scenario, there is no prior data for a particular type of damage. Even if there has been damage to a structure in the past, it is not likely that exactly the same type of damage will occur in the future, and thus the multi-class classification procedure described in Section 5.1 cannot be applied. On the other hand, data from an intact structure can be recorded easily. Damage detection then becomes a single-class classification problem, in which the goal is to detect whether new data sequences belong to the existing, intact case, or deviate from it and potentially indicate damage.

In the SSIM framework, as a benefit of the Bayesian approach, single-class classification can be simply reduced to multi-class classification by assuming that there are two classes ($K = 2$), that the first class indicates the intact scenario, and that there is no data for the second (damage) class ($\mathcal{Y}_2^{tr} = \emptyset$, $\mathcal{Z}_2^{tr} = \emptyset$, $N_2^{tr} = 0$). Equation 5.1 can now be written as:

$$\begin{aligned} &P(Z^{test} = k \,|\, \mathcal{Y}^{test}, \mathcal{Y}_1^{tr}, \mathcal{Z}_1^{tr}) \\ &\propto P(Z^{test} = k \,|\, \mathcal{Z}_1^{tr}) \cdot P(\mathcal{Y}^{test} \,|\, Z^{test} = k, \mathcal{Y}_k^{tr}, \mathcal{Z}_k^{tr}), \end{aligned} \tag{5.7}$$

where, from Equation 5.2,

$$\begin{aligned} P(Z^{test} = k \,|\, \mathcal{Z}_1^{tr}) &\equiv P_{tr}(Z^{test} = k) \\ &= \begin{cases} \frac{\alpha_1 + N_1^{tr}}{\alpha_1 + N_1^{tr} + \alpha_2} & , k = 1 \\ \frac{\alpha_2}{\alpha_1 + N_1^{tr} + \alpha_2} & , k = 2 \end{cases}, \end{aligned} \tag{5.8}$$

and, from Equation 5.3,

$$\begin{aligned} P(\mathcal{Y}^{test} \,|\, Z^{test} = k, \mathcal{Y}_k^{tr}, \mathcal{Z}_k^{tr}) &= \begin{cases} \mathscr{L}_1(\mathcal{Y}^{test} \,|\, \mathcal{Y}_1^{tr}) & , k = 1 \\ \mathscr{L}_2(\mathcal{Y}^{test}) & , k = 2 \end{cases} \\ &= \begin{cases} \sum_{\tilde{E}_1} \int_{\tilde{\theta}_1} P(\mathcal{Y}^{test} \,|\, \tilde{E}_1, \tilde{\theta}_1)\, P(\tilde{E}_1, \tilde{\theta}_1 \,|\, \mathcal{Y}_1^{tr})\, d\tilde{\theta}_1 & , k = 1 \\ \sum_{\tilde{E}_2} \int_{\tilde{\theta}_2} P(\mathcal{Y}^{test} \,|\, \tilde{E}_2, \tilde{\theta}_2)\, P(\tilde{E}_2, \tilde{\theta}_2)\, d\tilde{\theta}_2 & , k = 2 \end{cases}. \end{aligned} \tag{5.9}$$

Here, $P(\tilde{E}_2, \tilde{\theta}_2 \,|\, \mathcal{Y}_2^{tr} = \emptyset) = P(\tilde{E}_k, \tilde{\theta}_k)$ is simply the prior probability of structure and parameters for class 2 (damage scenario), while $\mathscr{L}_2(\mathcal{Y}^{test})$ is the marginal likelihood of the test sequence under that prior.

Finally, Equation 5.4 can now be specialized to:

$$\begin{aligned} &P(Z^{test} = 1 \,|\, \mathcal{Y}^{test}, \mathcal{Y}_1^{tr}, \mathcal{Z}_1^{tr}) \\ &\qquad = \frac{P_{tr}(Z^{test} = 1)\, \mathscr{L}_1(\mathcal{Y}^{test} \,|\, \mathcal{Y}_1^{tr})}{P_{tr}(Z^{test} = 1)\, \mathscr{L}_1(\mathcal{Y}^{test} \,|\, \mathcal{Y}_1^{tr}) + P_{tr}(Z^{test} = 2)\, \mathscr{L}_2(\mathcal{Y}^{test})} \end{aligned} \tag{5.10}$$

$$\begin{aligned} &P(Z^{test} = 2 \,|\, \mathcal{Y}^{test}, \mathcal{Y}_1^{tr}, \mathcal{Z}_1^{tr}) \\ &\qquad = \frac{P_{tr}(Z^{test} = 2)\, \mathscr{L}_2(\mathcal{Y}^{test})}{P_{tr}(Z^{test} = 1)\, \mathscr{L}_1(\mathcal{Y}^{test} \,|\, \mathcal{Y}_1^{tr}) + P_{tr}(Z^{test} = 2)\, \mathscr{L}_2(\mathcal{Y}^{test})}, \end{aligned}$$

which are the probabilities of a given test sequence being "intact" or "damaged", respectively. The higher values of $P(Z^{test} = 2 \mid \mathcal{Y}^{test}, \mathcal{Y}_1^{tr}, Z_1^{tr})$ mean that the dynamics of the test sequence deviates more from the dynamics of intact (training) sequences, which we relate to a higher probability of damage.

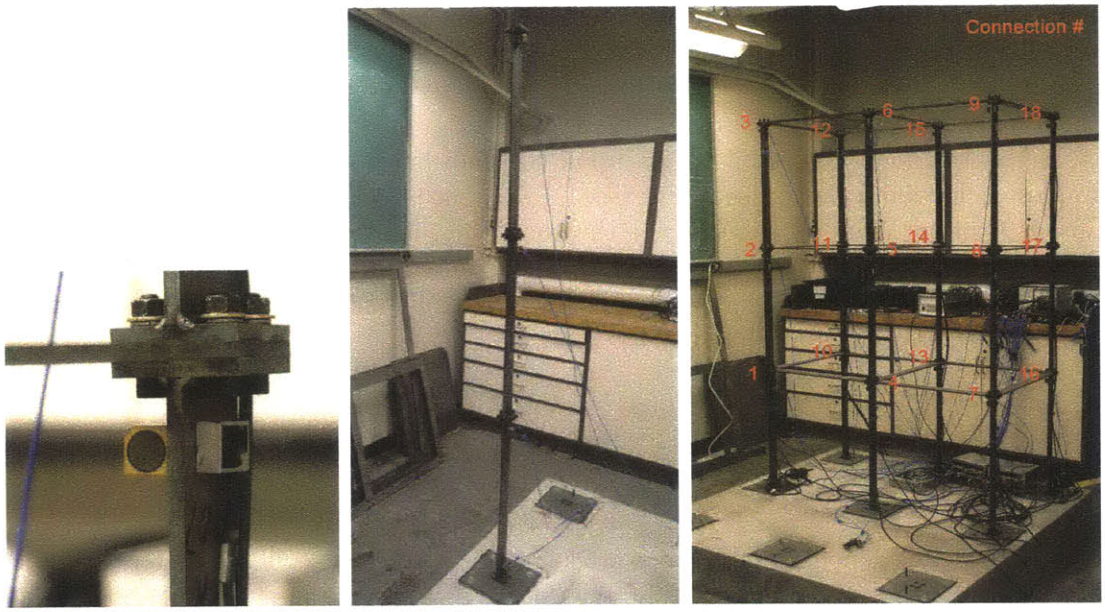
In practice, one may want to act upon the knowledge of damage probability. The simplest rule would be to use a threshold, ϵ_{dam}, such that further investigation is required if this probability exceeds the threshold, i.e., if $P(Z^{test} = 2 \mid \mathcal{Y}^{test}, \mathcal{Y}_1^{tr}, Z_1^{tr}) \geq \epsilon_{dam}$. A more sophisticated rule could be that different actions are taken for different levels of damage probability (i.e., when exceeding different thresholds). By rewriting the formula for damage probability as

$$P(Z^{test} = 2 \mid \mathcal{Y}^{test}, \mathcal{Y}_1^{tr}, Z_1^{tr}) = \frac{\frac{P_{tr}(Z^{test}=2)}{P_{tr}(Z^{test}=1)} \cdot \frac{\mathscr{L}_2(\mathcal{Y}^{test})}{\mathscr{L}_1(\mathcal{Y}^{test} \mid \mathcal{Y}_1^{tr})}}{1 + \frac{P_{tr}(Z^{test}=2)}{P_{tr}(Z^{test}=1)} \cdot \frac{\mathscr{L}_2(\mathcal{Y}^{test})}{\mathscr{L}_1(\mathcal{Y}^{test} \mid \mathcal{Y}_1^{tr})}}, \tag{5.11}$$

we can see that it depends on the ratio of likelihoods of the test sequence under the intact and prior models, $\frac{\mathscr{L}_2(\mathcal{Y}^{test})}{\mathscr{L}_1(\mathcal{Y}^{test} \mid \mathcal{Y}_1^{tr})}$, and on the ratio of damage and intact probabilities prior to seeing a test sequence, $\frac{P_{tr}(Z^{test}=2)}{P_{tr}(Z^{test}=1)}$. The first ratio may depend on the choice of the dependence model (e.g., linear Gaussian in LG-SSIM) and its hyperparameters (prior on structure and paramaters), but, assuming that these are appropriate/reasonable choices, it most importantly depends on the test sequence itself and how it differs from training sequences. On the other hand, the second ratio, $\frac{P_{tr}(Z^{test}=2)}{P_{tr}(Z^{test}=1)} = \frac{\alpha_2}{\alpha_1 + N_1^{tr}}$, depends only on the prior parameters α_1 and α_2 and on the number of training sequences. By controlling parameters α_1 and α_2, this ratio can be set to an arbitrary value (assuming fixed training data). Note that α_1 and α_2 are pseudo-counts of intact and damaged sequences that reflect our prior belief in the probability of intact versus damage scenario. Intuitively, one should expect a low probability of damage, and thus $\alpha_2 \ll \alpha_1$. On the other hand, the prior probability of damage can be set higher than expected (e.g., $\alpha_2 \approx \alpha_1$), which would reflect the "fear" of damage and increase the posterior probability that a test sequence belongs to a damage scenario. That would simply mean that a larger number of test sequences would "alarm" for damage. Note however the same effect could be achieved by decreasing the "alarm" threshold, ϵ_{dam}. Note also that, instead of using the posterior probability of damage, $P(Z^{test} = 2 \mid \mathcal{Y}^{test}, \mathcal{Y}_1^{tr}, Z_1^{tr})$, to indicate the possibility of damage, one can equivalently use the ratio of posterior probabilities of damage and intact scenarios:

$$\frac{P(Z^{test} = 2 \mid \mathcal{Y}^{test}, \mathcal{Y}_1^{tr}, Z_1^{tr})}{P(Z^{test} = 1 \mid \mathcal{Y}^{test}, \mathcal{Y}_1^{tr}, Z_1^{tr})} = \frac{P_{tr}(Z^{test} = 2)}{P_{tr}(Z^{test} = 1)} \cdot \frac{\mathscr{L}_2(\mathcal{Y}^{test})}{\mathscr{L}_1(\mathcal{Y}^{test} \mid \mathcal{Y}_1^{tr})}, \tag{5.12}$$

and devise rules based on the value of this ratio (e.g., ratio of 1 is equivalent to the damage probability of 0.5).



(a) Bolted connection (b) Simple column structure (c) 3 story 2 bay structure

Figure 5.3: Details of the laboratory setup

■ 5.3 Experiments with Laboratory Structures Data

Two experimental test structures were used to generate data to test the approach for application on a structure. Both structures are made of modular elements that are based on steel columns that are 60 cm × 5.08 cm × 0.64 cm, and bolted together by 4 bolts at each connection as shown in Fig. 5.3a as an example of a typical connection. The structures are bolted to a heavy concrete foundation as a reaction mass. They are instrumented with piezoelectric triaxial accelerometers that have a sampling rate of 6000 Hz, and the number used differs for each structure.

The first, simpler structure is a vertical cantilever beam that consists of three steel column elements shown in Fig. 5.3b. Damage is introduced on one of the two middle bolted connections in either a minor damage case where two of four bolts in the flexible direction are removed, or a major damage case where the four bolts are loosened to only be hand tight. This structure is instrumented with 4 accelerometers, one at each connection, including the connection with the foundation, and at the top of the structure. In order to excite the cantilever beam, it is displaced by approximately 5 cm and then released and allowed to freely vibrate for 10 seconds, during which data was collected. There are 10 test sequences for each damage scenario, and they are summarized in Table 5.1a.

The second structure is a 3 story 2 bay configuration with a footprint of 120 cm × 60 cm as shown in Fig. 5.3c. The structure consists of steel columns and beam

Table 5.1: Test cases and damage scenarios for structural models.

(a) Column structure

Test Case	Damage Scenario
1	Intact column
2	Minor damage, lower joint
3	Major damage, lower joint
4	Minor damage, upper joint
5	Major damage, upper joint

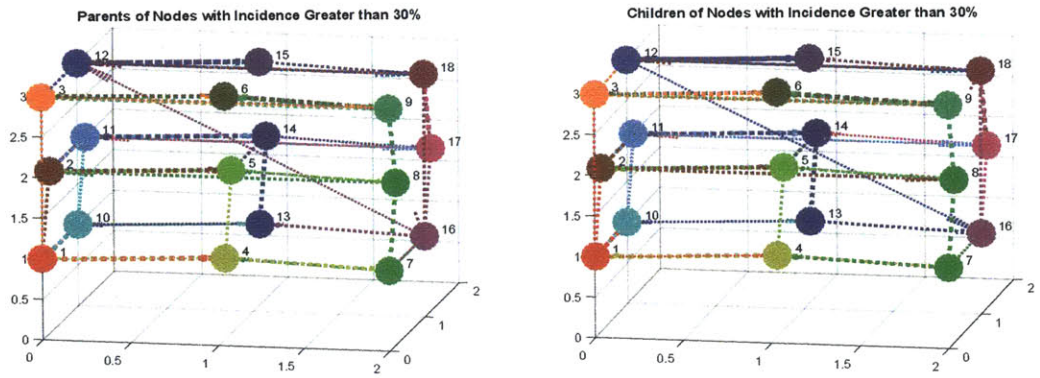
(b) 3 story 2 bay structure

Test Case	Damage Scenario
1	Intact column
2	Minor damage at 17
3	Major damage at 17
4	Minor damage at 1
5	Major damage at 1
6	Major damage at 1 and 17

frames of similar dimensions for each story that are bolted together to form each story. Damage is similarly introduced on the bolted connections with the minor and major damage cases by removing two bolts or loosening all four at connections 1 and 17, which are on opposite corners of the structure, with 1 being on the first story, and 17 being on the second. This structure is instrumented with 18 triaxial accelerometers at each of the connections between elements. For this structure the excitation is a small shaker with a weight of 0.91 kg and a piston weight of 0.17 kg that was attached to the top corner of the structure at connection 18, which provided a random white Gaussian noise excitation in the frequency range of 5 - 350 Hz in the flexible direction. Test measurements lasted for 30 seconds, during which the shaker is always exciting the structure, thus there is no ramp up or unforced section of the data. The damage scenarios are summarized in Table 5.1b. For each damage scenario, 10 sequences were acquired.

■ 5.3.1 Interaction Analysis

We analyze the results of inference over dependence structure among signals from different sensors on the **3-story 2-bay structure**. The number of parents of each node is bounded to 4, including the assumed self-dependency (therefore, 3 additional parents are allowed). Each data sequence is split into 18 subsequences that are 10,000 samples long. For each class, the posterior distribution over edges is computed on 180 subsequences that belong to that class (10 original sequences, 18 subsequences each) and then averaged out. The averaging is performed to get a stable result, since the posterior distribution fluctuates across subsequences. A visualization of the parent and child relationships for the intact structure is shown in Fig. 5.4. Colors represent the node the relationship originates from, and the width of the line represents the edge probability (wider is more likely). Specifically, relationships are plotted if their probability is higher than 0.3, and in Fig. 5.4a, the parents of the nodes are plotted, while in Fig. 5.4b the children of nodes are plotted. The nodes are vaguely arranged in the physical shape of the structure, and we can see that a lot of the same possible relationships in the physical structure, such as the columns, the beams, and the cross beams between the



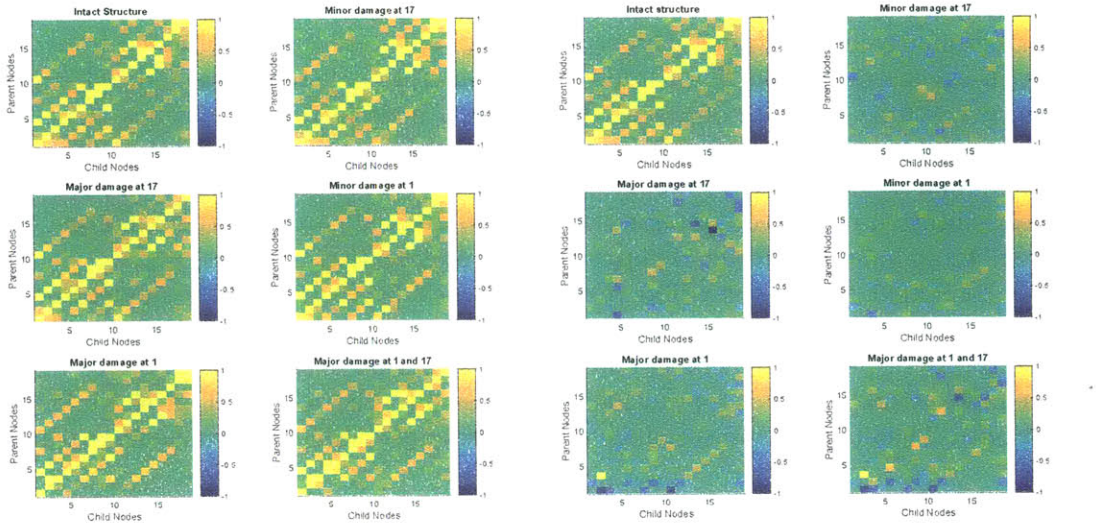
(a) Parents of Nodes (b) Children of Nodes

Figure 5.4: 3D Visualization of node parent and child relationships with probability above 0.3.

two sides of the structure, also show up in the inferred dependence structure.

Edge posteriors are also visualized as a matrix where the rows are the parents, and the columns are the child nodes, shown in Fig. 5.5a. We see that there are two quadrants where the relationships are strong, the 1-9 parent child relationships, and the 10-18, which correspond to the two sides of the structure. Within these quadrants, we see that there are strong relationships in groups of three, 1-3 for example, suggestive of the columns in the structure. We also see that there are relationships between nodes separated by three, such as 1 - 4 - 7, and similar for all the other nodes, which are suggestive of the beams that connect the nodes in the same story of the structure. Then, the other strong relationship is between the two sides of the structure, 1 - 10, 2 - 11, etc. which is seen as an off diagonal.

The results of inference for the other damage scenarios are also shown, and they mostly resemble the structure for the intact scenario. Looking at Fig. 5.5b instead, where for the damage cases, we show the difference from the intact scenario is shown, a couple of differences become more obvious. For both of the minor damage scenarios, the differences are minimal. However for major damage at node 1, we see that node 1 is now less likely a parent of nodes 2, 3, and 10. For example, the most likely parents of node 2 in the intact structure are nodes 1, 5 and 11, but for major damage at node 1, node 1 is replaced by node 3 on this list. Note that sensor 1 is actually slightly below the joint, so the damaged joint stands between nodes 1 and 2. For major damage at node 17, node 13 is much less often a parent of node 15, and the same for node 14, being a parent of node 13, all nodes that are physically close to node 17. Also, the dependence of node 18 on nodes 16 and 17 is reduced, as well as the dependence of node 17 on node 18. Note that the damaged joint stand between node 18 on one side and nodes 16 and 17 no the other side. Similarly, the dependence of node 11 on node 17, between which the shortest path goes through the damaged joint, becomes less likely. Finally, in the



(a) Structure for 4 Parent Nodes (b) Inferred Structure, Difference from Intact

Figure 5.5: Probability of parent nodes over many tests for intact and damaged cases.

dual major damage scenario at both 1 and 17, both these effects are seen in the inferred structure.

■ 5.3.2 Classification Results

We consider the problem of classification of sequences according to the structure condition, as described in Section 5.1. This problem is not directly applicable to real civil structures, as either damage has never occurred or it is unlikely that exactly the same damage scenario will occur in the future. However, it tells us how well the algorithm can distinguish not only damage from intact, but also different damage scenarios from each other. It is also worth noting that in some other damage detection problems, such as with machine parts, classification may actually be a realistic approach, as there may only be a handful of types of damages that typically occur and data from such scenarios may be available.

In each dataset, there are 10 sequences of each class. We perform 10 rounds of classification. In round j, sequence j from each class is included in the training set, while the other 9 sequences of each class are used for testing. Classification results are then averaged over all 10 rounds. To reduce computation, a subsequence of length 5,000 is used from each sequence, except in the experiments that test the effect of training and test sequence lengths. Although the results with longer sequences may be slightly better, they are not qualitatively different.

We employ a latent-AR LG-SSIM model for classification. We find that AR order

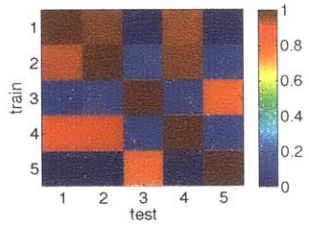
5 is sufficient to produce good classification result, although there is a slight advantage by further increasing this order. Hyperparameter values are either estimated from data or set in a general fashion (e.g., implying a broad prior distribution). In all experiments, we assume presence of a self edge for each node in the dependence structure. The bound on the number of additional allowed parents is set to 3 (maximum) in the single column case. In the 3 story 2 bay structure data, however, we found that the best classification results are obtained when no additional parents (other than self) are allowed. Explaining this result requires further investigation.

We compared the classification results obtained by the full SSIM model and an approximate model which assumes no observation noise (Section 5.1) and found that on the datasets presented here the full model performs only slightly better, but at the significant additional computational cost (mainly due to step 1 in the inference algorithm). Therefore, we present here detailed results obtained using the approximate model.

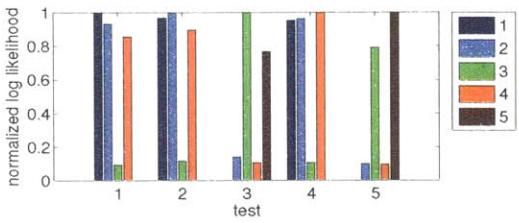
Single column structure results

First, for each pair of classes i and j, we compute the average log-likelihood of a test sequence from class i given a training sequence from class j (the average is over all pairs of sequences from classes i and j). Note that the average log-likelihoods do not account for the variability within a class and thus can only partially predict classification results. However, they can be considered as a measure of (asymmetric) similarity between classes. In particular, the comparison of log-likelihoods of a test class given different training classes is useful to indicate its possible "confusion" with other classes. The log domain is chosen to bring likelihoods closer to each other for the purpose of illusrtation, since the differences in likelihoods are huge in their original domain.

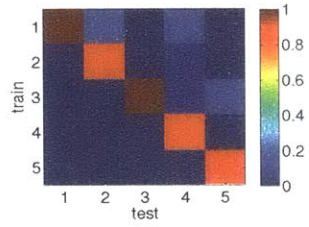
The resulting class-class log-likelihood matrix is shown in Fig. 5.6a. For the purpose of visualization, each column is normalized to contain values between 0 and 1, which does not change the relative comparison of values within a column. A different visualization of the same log-likelihood matrix is shown in Fig. 5.6b, in which each group of bars corresponds to a single test class, while bars within a group correspond to different training classes. Clearly, the average log-likelihood of each class is the highest when conditioned on sequences from the same class (diagonal entries). This suggests that the model indeed captures important features pertained to each class via posterior distribution of parameters. However, for some classes, the log-likelihood is also relatively high when conditioned on some of the classes other than itself. For example, the intact class (1) and the two minor damage classes (2 and 4) are the closest to each other in that sense. Also, the two major damage classes (3 and 5) are close to each other, although less than the previous three classes. On the other hand, there is a significantly higher separation between the low- and high-damage classes, and, as we will see next, a sequence from one of these groups is rarely misclassified as belonging to a class from the other group.



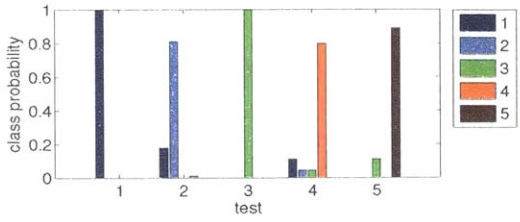
(a) log-likelihood matrix



(b) log-likelihoods grouped by test class



(c) classification frequencies matrix

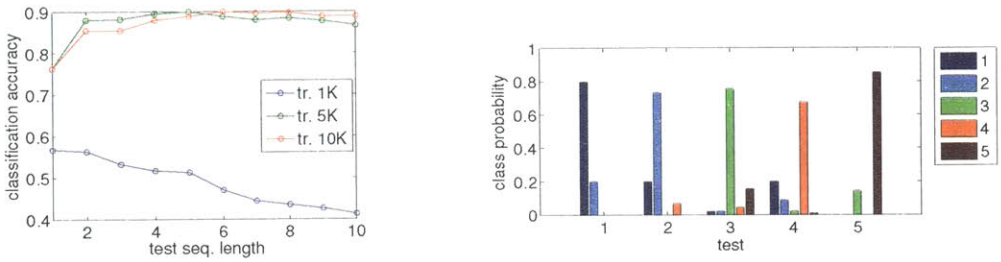


(d) classification frequencies grouped by test class

Figure 5.6: Column structure data class-class log-likelihoods are shown as (a) matrix and (b) bar groups. Similarly, classification frequencies are shown as (c) matrix and (d) bar groups.

Classification results are shown in Figs. 5.6c and 5.6d. Again, these are two different visualizations of the same results. For each pair of classes, test class i and training class j, the frequency of classifying a test sequence from class i as belonging to class j is shown. Therefore, each column in the matrix in Fig. 5.6c, as well as each group of bars in Fig. 5.6d, must sum to one. Overall, sequences are classified correctly most of the times (high diagonal values). Sequences from the two minor damage classes (2 and 4) are occasionally misclassified as belonging to the intact class (1), while sequences from the two major damage classes (3 and 5) are never misclassified as belonging to one of the low-damage classes and occasionally misclassified as belonging to the other major damage class.

Finally, we analyze classification accuracy as a function of training and test sequence lengths. Fig. 5.7a shows the overall classification accuracy (averaged over all classes) for three different training sequence lengths, $1,000$, $5,000$ and $10,000$, and ten test sequence lengths ranging from $1,000$ to $10,000$. Interestingly, for a fixed training sequence length, classification accuracy increases as the test sequence length increases only until it becomes equal to the training sequence length, after which it start decreasing. This result suggests that the properties of these time-series data change over time. Namely, subsequence for training and testing are always extracted starting at the same time in all sequences. Therefore, when training and test sequences are of the same length, they are aligned with respect to where they are in the measurement process (assuming that different sequences are measured under the same or very similar



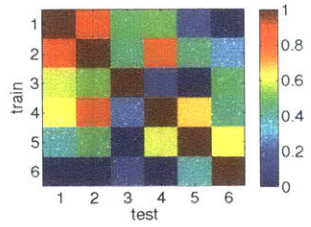
(a) (b)

Figure 5.7: (a) Overall classification accuracy on column structure data as a function of training and test sequence lengths. (b) Classification frequencies (by test class) when training and test sequence lengths are $5K$ and $1K$, respectively.

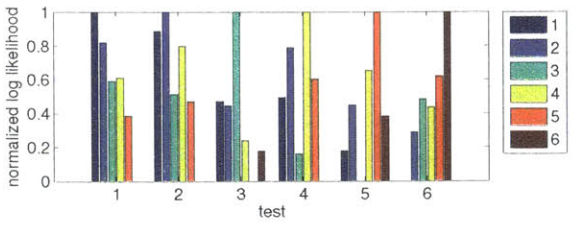
conditions). However, when the test sequence length increases beyond the training sequence length, test sequences start to increasingly incorporate parts of the process that was not included in training. Similarly, when test sequences are shorter than training sequences, training sequences include characteristics of a broader window of the process than is tested. This also can explain why the classification results are overall not better when the training sequence length is $10,000$ than when it is $5,000$. Likely, a window of $10,000$ is too broad and the additional amount of data, the second $5,000$ samples, does not help, since it differs in behavior than the first $5,000$ time samples. Naturally, there is a tradeoff between this behavior and the sequence length. For example, $1,000$ is too short, and the results with that length are clearly much worse. The phenomenon explained here could be attributed to the nature of excitation used in this setup, which is free vibration. The results with the shaker excitation, shown below, do not follow this pattern and behave as with one's expectations – more test or training data consistently yields higher accuracy. Lastly, Fig. 5.7b shows classification results for training and test sequence lengths equal to $5,000$ and $1,000$, respectively, which could be compared to the results in Fig. 5.6d, in which both lengths are $5,000$.

3-story 2-bay structure results

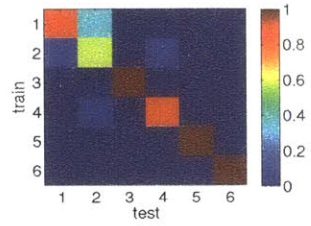
We present the same set of results on the 3-story 2-bay structure data. Average log-likelihoods between all pairs of classes are shown as a matrix in Fig. 5.8a and as bars grouped by test class in Fig. 5.8b. Again, these log-likelihoods are normalized such that each column in the matrix are between 0 and 1. As with the single column structure, the average log-likelihood of a sequence of one class is the highest when conditioned on a sequence from that same class (diagonal elements), and the highest confusion is between the low-damage classes, namely, the intact class, 1, and the two minor damage classes, 2 and 4. The lesser major damage classes, 3 and 5, seem to be occasionally confused as classes with either smaller or higher damage relative to them. Finally, the greater major damage class, 6, is most similar to the lesser major damage classes.



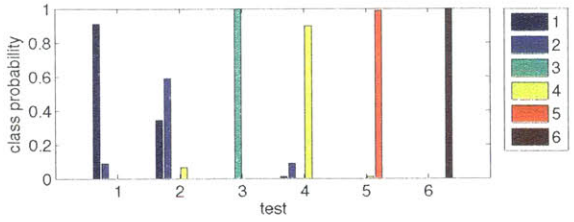
(a) log-likelihood matrix



(b) log-likelihoods grouped by test class



(c) classification frequencies matrix

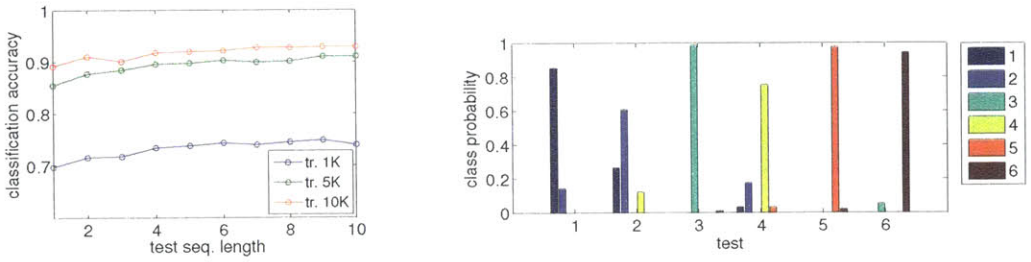


(d) classification frequencies grouped by test class

Figure 5.8: 3 story 2 bay structure data class-class log-likelihoods are shown as (a) matrix and (b) bar groups. Similarly, classification frequencies are shown as (c) matrix and (d) bar groups.

Classification results in terms of frequencies (fraction of times a sequence from one class is classified as belonging to another class) are shown as a matrix in Fig. 5.8c and as bars grouped by test class in Fig. 5.8d. Sequences from major damage classes (3, 5 and 6) are classified almost perfectly. On the other hand, some confusion between the three low-damage classes (1, 2 and 4) is present. In particular, sequences from the class that corresponds to a minor damage at node 17 are often misclassified as belonging to the intact class. This could possibly be attributed to the closeness of this node to the shaker.

The overall classification accuracy as a function of training and test sequence lengths is shown in Fig. 5.9a. Three different training sequence lengths were used, $1,000$, $5,000$ and $10,000$, while the test sequence length is varied from $1,000$ to $10,000$. Unlike with the single column structure results, classification accuracy on the 3 story 2 bay structure data consistently improves with the increased length of either training or a test sequence. This trend suggests that there is likely no significant variability in the dynamics of a sequence over time, and, consequently, longer sequences represent effectively more data. This is an expected behavior, since excitation provided by the shaker is uniform over time. Finally, for comparison with the results in Fig. 5.8d, in which both lengths are $5,000$, Fig. 5.9b shows classification results when training and test sequence lengths are $5,000$ and $1,000$.



(a) (b)

Figure 5.9: (a) Overall classification accuracy on 3 story 2 bay structure data as a function of training and test sequence lengths. (b) Classification frequencies when training and test sequence lengths are $5K$ and $1K$, respectively.

■ 5.3.3 Single-Class Classification Results

We evaluate the performance of single-class classification only on the **3-story 2-bay structure** data, as it presents a more challenging case than the single column structure data. As in the evaluation of classification above, subsequences of length 5,000 are used for training and testing. For each training and each test sequence, the value $\frac{\mathscr{L}_2(\mathcal{Y}^{test})}{\mathscr{L}_1(\mathcal{Y}^{test}|\mathcal{Y}_1^{tr})}$ is computed, where labels 1 and 2 correspond to intact and damage classes, as in Section 5.2. The test sequence is classified as anomalous if this value is above some threshold ϵ_{dam}. Note that this is equivalent to using Equation 5.12, since the ratio $\frac{P_{tr}(Z^{test}=2)}{P_{tr}(Z^{test}=1)}$ is determined by the prior and can be absorbed into the threshold.

ROC curve [38], which represents the rate of true positives as a function of the rate of false positives, is computed separately for each damage scenario by varying the value of the threshold ϵ_{dam}. Cross-validation is used to increase the number of training-test pairs. In each round, one sequence from intact scenario is considered as a training sequence, while the remaining 9 intact sequences and all 10 sequences from the chosen damage scenario are treated as test sequences. The number of false positives and the number true positives are computed as a function of ϵ_{dam} and aggregated over all rounds (i.e., over all choices of a training sequence).

Thus computed ROC curves for all damage classes are shown in Figure 5.10. ROC curves are "perfect" for all major damage scenarios, in that there is a threshold for which all test sequences are correctly classified (i.e., the value $\frac{\mathscr{L}_2(\mathcal{Y}^{test})}{\mathscr{L}_1(\mathcal{Y}^{test}|\mathcal{Y}_1^{tr})}$ is below the threshold for all intact test sequences and it is above the threshold for all sequences from the damage case). Note that the ROC curves for scenarios 3 and 5 are not visible in Figure 5.10 because they are overlayed by the curve for scenario 6. The ROC curve for the case of minor damage at node 1 (scenario 4) is close to perfect, while the worst result is for the case of minor damage at mode 17 (scenario 2). This is not surprising, given that the we already found in the previous section that most errors in a classification

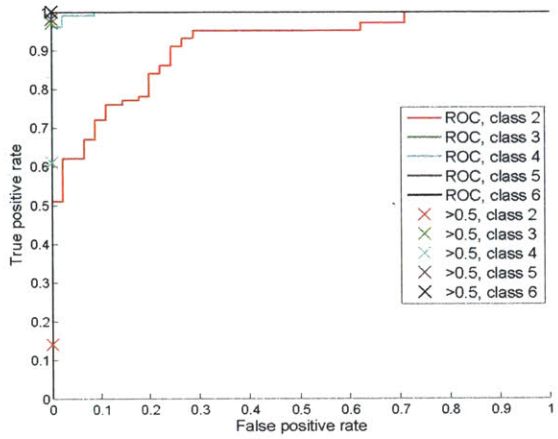


Figure 5.10: ROC curves for each damage scenario on 3-story 2-bay structure data. Points on the curves that correspond to the posterior probability of damage equal to 0.5 are marked with an 'x'.

setting occur when an intact sequence is misclassified as belonging to scenario 2, and vice versa, i..e, that sequences from these two classes are most similar to each other.

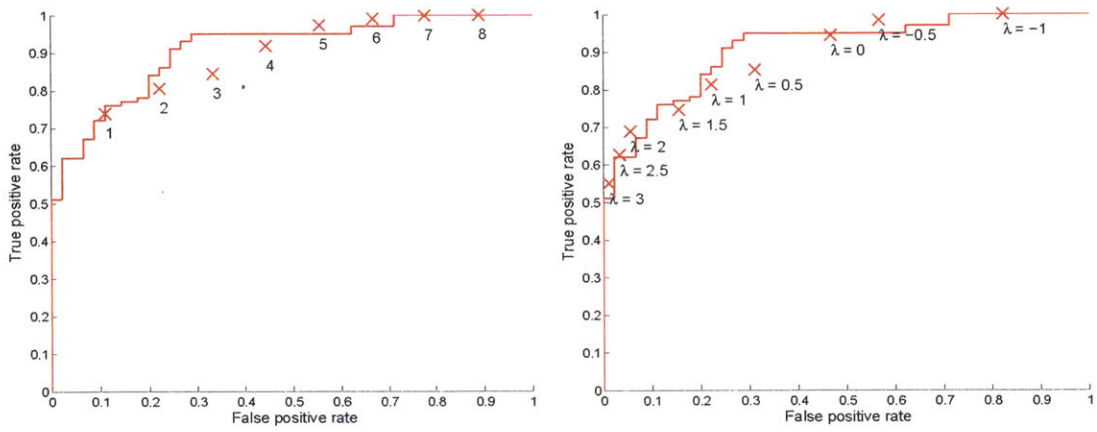
In addition, for each damage scenario, a point that corresponds to the threshold value $\epsilon_{dam} = 1$ is shown in Figure 5.10 with an 'x' mark. This threshold value corresponds to the posterior probability of a test sequence being damaged equal to 0.5, under the assumption that the prior probabilities of a test sequence being intact or damaged are equal. Note that there are no false positives in any of the scenarios. In other words, the posterior probability that a sequence is damages is never higher than 0.5 for intact sequences. On the other hand, for the major damage scenarios, this probability is above 0.5 for almost all damaged sequences. However, only about 60% of damaged sequences have posterior probability of damage above 0.5 in case of the minor damage at node 1, and less than 15% of damaged sequences are classified as damaged by this rule in the case of the minor damage at node 17.

If one wants to devise a threshold rule in practice, the threshold that corresponds to the posterior probability of damage of 0.5 is not necessarily the right choice. From Figure 5.10 we can see that, in the case of minor damages, this rule would classify a sequence as damaged only when it's very certain of it. If one wants to be less conservative and detect more damaged cases (at the expense of false positives), the threshold should be set to a lower value. However, choosing that number may not be as intuitive as one may expect. The likelihood of a sequence depends on its length approximately exponentially since the likelihoods of variables at each time point are multiplied together.[2] As the length of a sequence increases, its likelihood quickly converges to either

[2]Technically, this is the case for a specific value of model structure and parameters, and the overall likelihood is obtained by summing/integrating over possible values of structures and parameters, weighted by their prior.

0 or 1. Similarly, the ratio $\frac{\mathscr{L}_2(\mathcal{Y}^{test})}{\mathscr{L}_1(\mathcal{Y}^{test}|\mathcal{Y}_1^{tr})}$, which is used to discriminate damaged from intact sequences, approaches 0 or 1 exponentially with sequence length, depending on whether the test sequence is more likely under the prior model or under the posterior model given the training sequence. In an ideal case, if the model perfectly matches the data, one could simply "trust" these probabilities – i.e., if the model tells that the probability of damage is 1, that would indeed mean that there is almost certainly damage, and, similarly, sequences with posterior probability of damage close to 0 would almost certainly correspond to an intact structure. However, due to the fact that the statistical model is only an approximation to the physical model, some sequences from a damage scenario may actually have low posterior probability of damage or some sequences from intact scenario may have high probability of damage. From the results above, we see that the former is the case for the 3-story 2-bay structure data.

One approach to compensate for the effects of sequence length and model mismatch is to adjust the threshold to account for them. However, that is a very hard problem, as it is difficult to quantify these effects precisely (or even approximately). Instead, we take a data driven approach to choosing a threshold. Since the assumption is that the data from a damage scenario is not available a priori, we can only use the data from the intact scenario. Specifically, we assume that one intact sequences is used as a training sequence, 8 intact sequences are used for tuning, and the remaining intact sequence is used for testing (along with all ten sequences from a damage scenario that is tested). First, the value of the ratio $\frac{\mathscr{L}_2(\mathcal{Y}^{test})}{\mathscr{L}_1(\mathcal{Y}^{test}|\mathcal{Y}_1^{tr})}$ is computed for all tuning sequences. Let $L_1^{tune}, \ldots, L_8^{tune}$ denote these values. A threshold is computed as a function of these values, which is then applied to classify the test sequences. This is repeated for all possible choices of a training sequence and tuning sequences among intact data, and the results are aggregated (which we refer to as "cross-validation" in this context). It remains to discuss how to choose the threshold ϵ_{dam} as a function of values $L_1^{tune}, \ldots, L_8^{tune}$. One possibility is to use the maximum of these values, which would result in low false positive rates, and, if the damage sequences are relatively different from intact sequences, would result in a large true positive rate. More generally, if these values are sorted such that $L_1^{tune} > L_2^{tune} > \ldots > L_8^{tune}$, then, choosing a threshold that is between i^{th} and $(i+1)^{st}$ value would approximately result in the false positive rate of $i/8$. Therefore, the false positives rate can be controlled even though the corresponding rate of true positives is not known a priori. Another approach is to assume that these value come from a Gaussian distribution and compute their empirical mean and standard deviation. The threshold can then be set as $\mathrm{E}L_i^{tune} + \lambda\sigma L_i^{tune}$, for some value of λ (e.g., $\lambda = 2$ would correspond to taking two standard deviations away from the mean). Figure 5.11 shows the tradeoff between the rates of true positives and false positives for these two approaches in the case of minor damage at node 17 (scenario 2). Figure 5.11a shows the tradeoff points when the threshold is set to $L_1^{tune}, \ldots, L_8^{tune}$, respectively, assuming that these values are sorted in the decreasing order. Figure 5.11b shows the tradeoff points for various values of λ when he threshold is set to $\mathrm{E}L_i^{tune} + \lambda\sigma L_i^{tune}$. Note that the points in both figures do not necessarily fall on



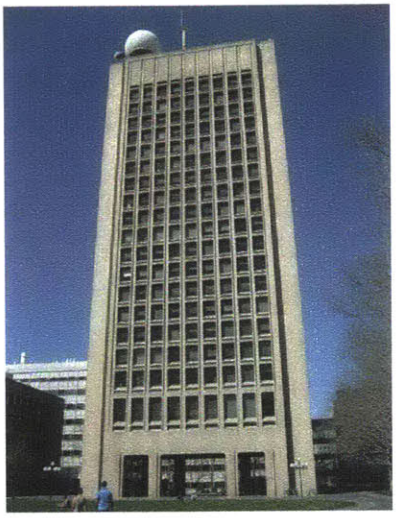
(a) (b)

Figure 5.11: Points of tradeoff between the rates of true positives and false positives when: (a) The threshold is set to $L_1^{tune} > L_2^{tune} > \ldots > L_8^{tune}$, repsectively. (b) The threshold is set to $\mathrm{E}L_i^{tune} + \lambda\sigma L_i^{tune}$ for different values of λ.

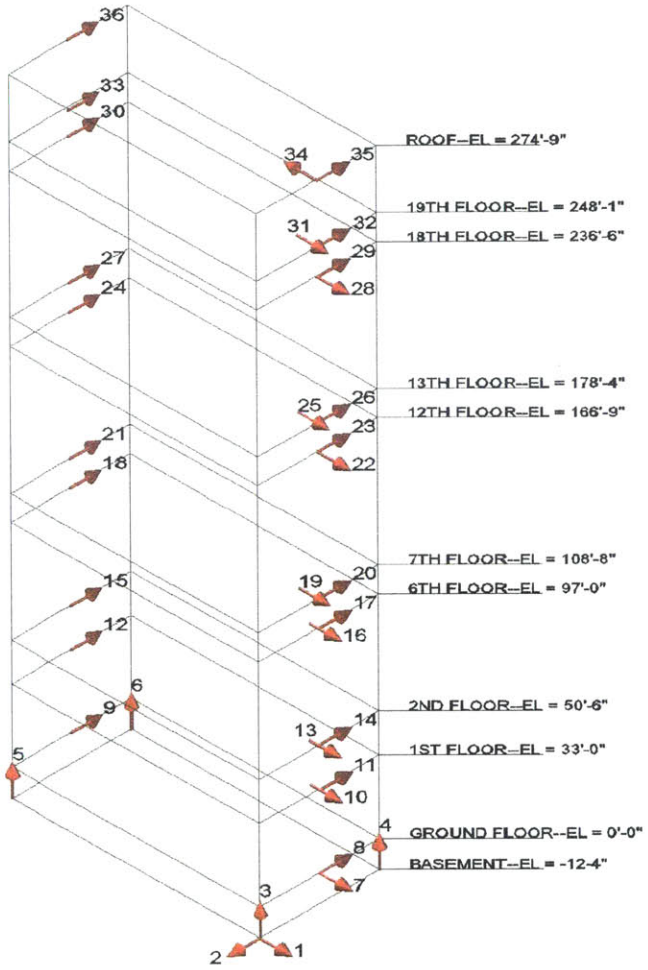
the ROC curve because thresholds are a function of tuning sets and are therefore not necessarily uniform across all training-tuning sets.

■ 5.4 Experiments with Green Building Data

The Green Building is a 21 story building on the campus of the Massachusetts Institute of Technology that has been instrumented by an accelerometer system, used as a testbed for system identification and structural health monitoring studies [9]. The building itself is shown in Fig. 5.12a and the locations and directions of the 36 uniaxial accelerometers are shown in Fig. 5.12b. Data from these accelerometers was used to test the methodology in a different situation from the experimental structure, where there is no known damage or change in the structure between the different data collections. Instead, the excitation and environmental conditions for the structure vary greatly. They are summarized in Table 5.2. The excitation conditions vary from typical ambient vibrations, to a day with 20 mph sustained winds, to a 4.0 magnitude earthquake located approximately 100 miles away. The measurements were made in the months of April to October, and with air temperatures typical of Spring, Summer and Fall, with temperature effects potentially inducing small changes in the structure due to internal stresses from differential thermal expansion of materials. The goal with processing this data is to use the ambient excitation data as a baseline for the structure and detect when an anomalous event or excitation occurs, while not triggering false positives during similar ambient excitations, while under different environmental conditions. We subdivided the test cases into several sequences of 30,000 sample length. Some of the sequences



(a) MIT Green Building



(b) Sensor Locations

Figure 5.12: MIT Green Building

are longer than the others, so there are multiple sequences for some of the test cases. The test cases belonging to each excitation and/or environmental condition are given.

■ 5.4.1 Interaction Analysis

We use a subsequence of length 10,000 from the 6/22/2012 ambient recording to infer the dependence structure among sensor signals from the Green Building. An AR order of 5 was used, with a maximum of 3 additional parents allowed. We plot a visualization of the parent and child relationships in Fig. 5.13. The color in these plots shows the direction of the sensor in the building, with red for E-W, blue for N-S, and green for

Table 5.2: MIT Green Building Conditions.

Test Case	Date	Excitation/Condition
1	5/14/2012	Unknown Event
2-3	6/22/2012	Ambient
4-6	7/4/2012	Fireworks
7	10/16/2012	Earthquake
8-10	4/15/2013	Ambient
11-16	10/07/2013	Windy Day

vertical sensors. We see that there are many relationships between the sensors in the same direction, and fewer between sensors in different directions. Most relationships are between the sensors that are located close to each other. There is also a fair number of relationships across the structure for the NS sensors.

A particularly interesting observation is the lack of relationships between the vertical sensors except for the pairs of 3-4 and 5-6. This may be explained by the rocking behavior found in the building [9], where sensors 3 and 4 move in phase, in opposition to sensors 5 and 6.

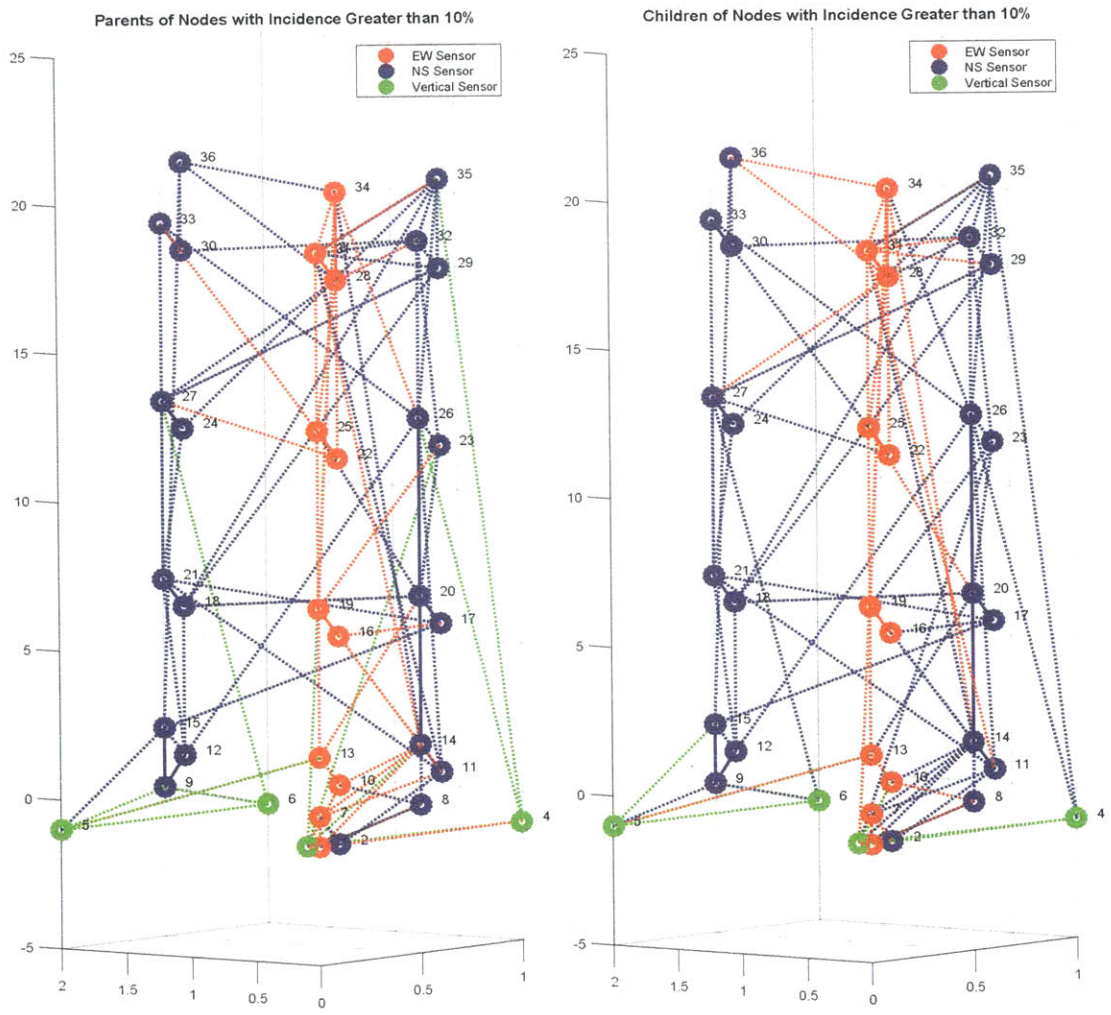
These relationships are also visualized in a matrix shown in Fig. 5.14. The sensors are grouped into vertical sensors, EW sensors, and then NS sensors, as given in the axis labels.

■ 5.4.2 Single-Class Classification

Fig. 5.15 shows the matrix of the logarithm of likelihood ratios, $\log \frac{\mathscr{L}_2(\mathcal{Y}^{test})}{\mathscr{L}_1(\mathcal{Y}^{test} \mid \mathcal{Y}_1^{tr})}$, normalized to be between 0 and 1 for the visualization purpose. The value at row i and column j corresponds to the ratio computed when sequence i is considered as a training sequence and sequence j as a test sequence. Recall from Equation 5.12 that this ratio can be used to discriminate sequences that behave differently from the training sequence. The higher the value of the ratio is, the more likely it is that the test sequence will be labeled differently from the training sequence.

We can see that the events that are the most similar to each other are the events in ambient conditions, windy conditions, but also the first two sequences for the fireworks event. For the fireworks event, when the recording was made during the Boston July 4^{th} fireworks show, only the last sequence of the three occurs during when the fireworks are being set off. The first two sequences are of the normal ambient structure, and thus they have low likelihood ratio with respect to the other ambient structure test cases. The windy condition measurements are not as dissimilar from the ambient measurements as we would have expected as winds were sustained at 20mph with gusts at higher speeds. The accelerations measured however are likely similar to ambient conditions with slightly higher magnitudes, as the winds are random excitations.

The last sequence in the fireworks test case, the earthquake, and the 5/14/2012 event



(a) Parents of Nodes

(b) Children of Nodes

Figure 5.13: 3D Visualization of Green Building node parent and child relationships with incidence over 10%

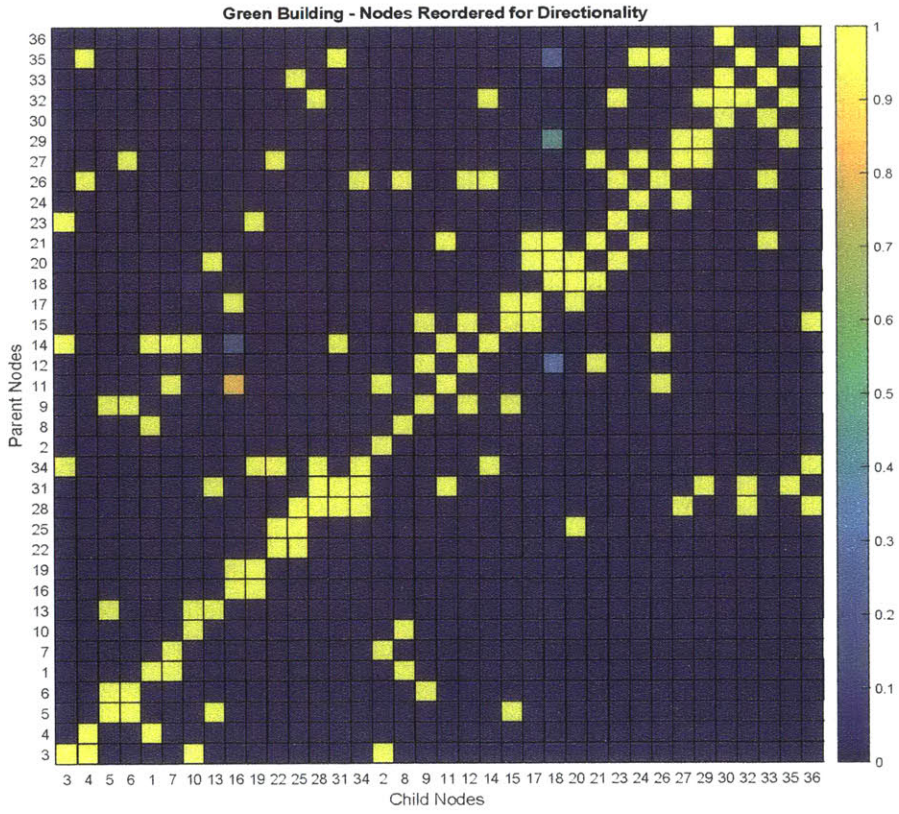


Figure 5.14: Matrix visualization of node incidence for Green Building. The sensors are grouped into vertical sensors, EW sensors, and then NS sensors, as given in the axis labels. Concentration of high probability edges around the diagonal shows that many relationships are between the sensors in the same direction and close to each other.

test cases all have significantly higher likelihood ratios with respect to the ambient cases. What's interesting is that the fireworks sequence is similar to the 5/14/2012 event, but both are dissimilar to the earthquake case. The 5/14/2012 event was when the recording system was triggered to record because accelerations exceeded a preset threshold, however there is no known event that corresponds to it. The time-series looks like a single impulse, possibly suggesting similar behavior induced in the structure to the series of impulses from the fireworks sequences. The third fireworks sequence seems to be the most dissimilar from all the other sequences.

What these results tell us is that we can likely classify when the structure has been excited in a significantly different way than typical ambient conditions. The differences between random ambient excitations and impulse excitations or earthquake excitations are clearly visible. We do not evaluate the performance of the single-class classification formally, as we did with the laboratory data using ROC curves, since the number of recorded sequences for the Green building is not that large. However, it is clear from the likelihood ratio matrix in Figure 5.15 that using any of the ambient sequences as a training sequence and a reasonable threshold rule (e.g., use other ambient sequences as tuning data and take the highest ratio among them as a threshold) would perfectly classify the sequences from the earthquake and the 5/14/2012 event and the third fireworks sequence as non-ambient. Sequences from the windy condition would also be classified as non-ambient in most cases, except when the second sequence of the 6/22/2012 ambient recording is used for training. In that case, the sequences from the windy condition have lower likelihood ratio than those from the 4/15/2013 ambient event. If the latter ones are used for tuning, the former ones would be classified as ambient. Also, note that the two ambient recordings are slightly different from each other, which could possibly be attributed to the temperature difference of 40°F between these two recordings. This suggests that acquiring more ambient recordings over time and in different conditions would be useful to understand the variation in them and how that relates to the classification problem.

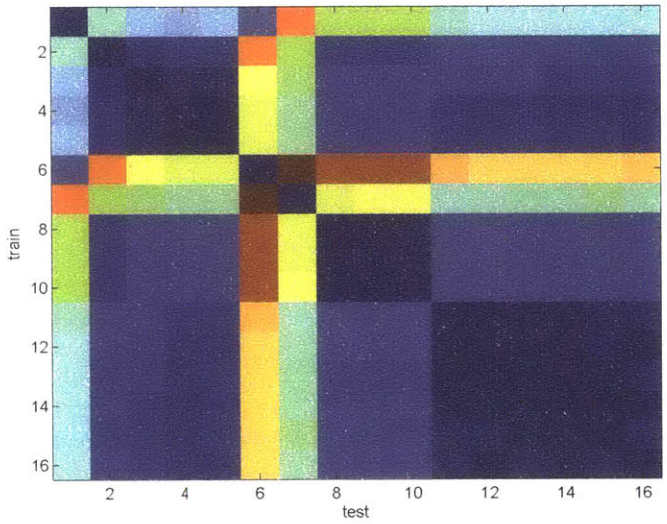


Figure 5.15: Matrix of the log-likelihood ratios, $\log \frac{\mathscr{L}_2(\mathcal{Y}^{test})}{\mathscr{L}_1(\mathcal{Y}^{test}|\mathcal{Y}_1^{tr})}$, between Green Building data sequences, normalized to be between 0 and 1. The value at row i and column j corresponds to the ratio computed when sequence i is considered as a training sequence and sequence j as a test sequence. The correspondence between sequence indices and events is: 5/14/2012 Unknown Event (1), 6/22/2012 Ambient Event (2-3), Fireworks (4-6), Earthquake (7), 4/15/2013 Ambient Event (8-10), and Windy Day (11-16). Note that the events that are the most similar to each other are the events in ambient conditions, windy conditions, but also the first two sequences for the fireworks event, which were recorded before the fireworks actually started. On the other hand, the last sequence in the fireworks test case, the earthquake, and the 5/14/2012 event test cases all have significantly higher likelihood ratios with respect to the ambient cases. These results suggest that we can likely classify when the structure has been excited in a significantly different way than typical ambient conditions.

Chapter 6

Conclusion

We presented a state-space switching interaction model (SSIM), which represents interactions as directed edges of a dynamic Bayesian network, allows switching between interactions, and allows arbitrary observation processes and missing data. Furthermore, we employed Bayesian reasoning over structures to deal with uncertainty in the data and due to the large number of possible structures. Efficient inference is enabled by limiting the number of parents per signal, and is done via a Gibbs sampling procedure. This model is expressive and can uncover different aspects of interactions among time-series and their patterns, as we have demonstrated by experiments. In addition, we developed a classification and a single-classification variants of the SSIM and showed that these models can be successfully applied to the problem of damage detection in civil structures.

6.1 Summary of Contributions

Modeling

We develop the SSIM framework for Bayesian inference over switching time-series interaction structure under uncertainty, which extends the work of Siracusa and Fisher [49, 50] by allowing for noisy and missing observations of time-series. We introduce a linear Gaussian SSIM model (LG-SSIM), in which both dynamics and observation models are linear Gaussian models, thus extending Gaussian state-space switching models to include structural inference. We also introduce a latent-AR variant of the LG-SSIM, in which an autoregressive (AR) model of an arbitrary order is allowed among the latent state variables. Both LG-SSIM and latent-AR LG-SSIM can be paralleled to analogous extensions of the model of Siracusa and Fisher [49, 50], in which direct observations of time-series are assumed.

Algorithms

We develop a Gibbs sampling procedure for inference in SSIM, which simultaneously reasons over interaction structures and parameters, the pattern of switching between different interactions, latent states associated with time-series, and observation model parameters. The algorithm extends the Gibbs sampling inference procedure of Siracusa

and Fisher [49, 50] to include steps in which latent states and observation model parameters are sampled. We also develop a specialization of the inference procedure for the LG-SSIM. In particular, we develop a numerically stable algorithm for block-sampling of latent states trajectories given observations that could be noisy and missing, and for dynamic models that allow for deterministic dependencies among state variables, such as in latent-AR LG-SSIM. In addition, we provide in-depth time and memory complexity analysis of the Gibbs sampling inference algorithm for the LG-SSIM. Finally, we provide guidelines for setting the prior (i.e., hyperparameters) in the LG-SSIM model, initializing latent variables, and performing a Gibbs sampling procedure. We also provide a procedure for evaluating a posterior distribution over a huge number of structures given a limited (much smaller) number of posterior samples obtained by the Gibbs sampling inference procedure.

Experiments

We use synthetic data to demonstrate the advantage of interaction analysis over testing pairwise relationships, and the advantage of the SSIM model over the model of Siracusa and Fisher [49, 50], which does not account for observation noise. We introduce a novel dataset, the joystick data, which is created specifically for testing results of interaction analysis in realistic conditions. It is developed in such a way that ground truth interactions are known by design, but it is human-generated and not synthesized from the model. We demonstrate the ability of the SSIM model to infer interactions and a switching pattern even in the presence of relatively high observation noise or if a significant fraction of data is missing, and that it is advantageous over the STIM model of Siracusa and Fisher [49, 50], as the STIM model does not handle missing data and performs worse in the presence of high observation noise. We also demonstrate the advantage of reasoning over structure posterior over MAP estimation, as spurious edges in a MAP structure estimate are typically assigned higher uncertainty (lower probability) in the posterior than the correct edges. Finally, we apply the SSIM model to a real-world problems and show types of analyses that it enables.

Structural Health Monitoring

We develop variants of the SSIM model for classification and single-class classification of time-series. On data from two laboratory model structures, we demonstrate that the SSIM classification model can classify time-series obtained under intact and different damage scenarios with high accuracy, in both standard and single-class classification settings. On the MIT Green building data, we demonstrate that the SSIM single-class classification model can distinguish time-series obtained under conditions that differ form ambient conditions (from those obtained under ambient conditions) and that it also predicts the "strength of deviation". We also perform interaction analysis on both datasets and show that inferred edges correlate with an actual physical structure.

6.2 Future Directions

We demonstrate the utility of the SSIM model for switching interaction analysis on two real-world examples: learning interactions among climate indices and among sensor data from civil structures. In addition, we apply the model to the problem of damage detection in civil engineering. However, there are many domain where the methodology developed her could be applied, such as finance / econometrics, social networks, neuroscience, health monitoring, transportation / traffic analysis, video analysis, sports / games, etc. In addition to numerous possible applications, the model can be extended or improved in various ways.

6.2.1 Scalable inference

Efficient inference over interaction structures in the SSIM model is enabled by using a modular prior on structure with additional constraints, such as a bounded in-degree constraint. While this approach significantly reduces the complexity of inference over interactions – from super-exponential to polynomial, it is still not efficient enough for applications on a very large number of signals. Further approximations to the model and/or different approximate inference algorithms are needed to improve the scalability of the approach.

For example, in step 4 of Algorithm 3.1, full parameter updates are computed prior to drawing a sample of dependence models. However, that may be avoided by developing an MCMC algorithm for sampling dependence models (step 4) that is integrated within the overall sampling algorithm. If the acceptance ratio is sufficiently high and the algorithm traverses the posterior space of structures efficiently (e.g., by including "jump" moves), such approach may be more efficient than exact sampling form the full structure posterior. Full posterior distribution over structure could still be computed in those sampling rounds from which a sample is extracted (as in Section 4.1.3), but that is only a fraction of times (e.g., every 50^{th} sampling round).

6.2.2 Nonparametric approaches

Currently, the SSIM model assumes that the number of possible switching states is known in advance. Although we showed (at least in one example) that by marginalizing the switching sequence, the number of visually distinct inferred structures can be correct even though the number of switching states is set incorrectly, setting the number of switching states properly yields better results and is in general beneficial. While the number of switching states can sometimes be guessed, it is often not the case. If that number is not known, one approach is to perform inference with different numbers of states and analyze the results to see which one best fits the data. Clearly, having an algorithm that automatically infers the number of switching states would be advantageous. That can be done using a Bayesian nonparametric approach. In particular, a sticky HDP-HMM model [18] of the switching state sequence can be used, which is a Markov model of a sequence with possibly infinite number of states, paired with a

hierarchical Dirichlet process prior [55] as a conjugate prior, and which also encourages persistence of states over time. The number of inferred states is then a function of data.

The SSIM framework assumes no relationship between dependence models of different signals, as well as those that pertain to different switching states. However, the same pattern of interaction may repeat with different signals and in different regimes. For example, the interaction "follow" appears in the joystick data between different pairs of players in different assignments. Hierarchical nonparametric Bayesian methods can be used to model "template" dependencies that are shared among different combinations of signals and their parents across different switching states. Since the parameters of the same type of dependency may vary to some degree when different signals are involved, the hierarchical approach is suitable for modeling both the base distribution of parameters and variations pertinent to different sets of signals. Also, a nonparameteric approach is advantageous since the number of possible template interactions is typically not known in advance. This approach is also applicable (and likely inevitable) in scenarios in which objects appear and disappear from the scene frequently, and the only hope to learn their behavior from data is that there might be a small number of patterns that repeat for different objects. An example application is traffic monitoring at an intersection: vehicles change all the time and their number is different, but there is a limited number of scenarios that may occur.

■ 6.2.3 Online learning

Many time-series data is constantly or periodically being collected, such as stock prices, sensor data in structural health monitoring, climate data, etc. In addition, inferring changes in interaction as soon as possible is very important in some domains. Therefore, developing algorithms that can efficiently update existing and learn new interaction models (and, in general, update the results of inference) with newly arrived data is important. However, exact inference with new data requires repeating the full inference procedure over all data because the posterior distribution of latent variables related to the "old" time points may change given new evidence. In other words, new data may influence our belief in the interaction structures and a switching pattern in the past. On the other hand, one may expect that after sufficient amount of data is seen pertaining to each interaction, the belief in that interaction should not change significantly. That can be exploited to develop approaches that perform joint inference over the new sequence of data and only selected time points from the past for which there is still significant uncertainty in the interaction structure and the switching state.

■ 6.2.4 Multi-scale interaction analysis

The SSIM model allows arbitrary orders of the AR model of latent time-series. However, the computational complexity of the inference algorithm step in which latent time-series states are sampled grows cubically with the order of the AR model (Table 3.2), Therefore there is a limit to the order of the AR model that can be used in practice (e.g., up to 100), and it may be difficult to infer long-range dependencies. On the other hand, long-

range dependencies are often among coarser versions of signals rather than the original signals. For example, if a person follows another person from a close distance, he may react quickly and to even small changes in behavior of the person he follows. On the other hand, if he follows the other person from far away, he may only react to large changes in behavior of the other person, and with a larger delay.

We refer to longer-range dependencies among coarser versions of signals as dependencies at a coarse scale. Note that one (but not the only) way to define different signal scales is to remove high frequency content and leave only frequencies up to some threshold. The lower the threshold frequency is, the coarser the signal is. If coarse versions of signals are down-sampled, long-range dependencies among the original signals become short-range dependencies among down-sampled coarse signals. A possible way to learn such dependencies is therefore to apply a low-pass filter to original signals, down-sample them, and then perform interaction analysis on thus obtained signals using the SSIM model. Note that this approach decouples interaction analysis at different scales, as input signals are processed independently for each scale. That is fine when signals are observed directly. However, if signals are observed through a noisy process, decoupling inference at different scales may result in inferring different latent time-series that correspond to a same signal. Furthermore, the method would require some way of dealing with missing data that may not be principled. Developing a generative multi-scale model of signals that incorporate interactions among signals would allow for a joint inference over interactions at different scales and would deal with observation noise and missing data in a principled way, as in the SSIM model.

■ 6.3 Final Thoughts

Understanding interactions is an important question in many domains, but learning interactions from data remains a challenging problem. This thesis attempts to extend the arsenal of tools for tackling this problem by developing a method for efficient Bayesian inference over switching temporal interaction structure from noisy data. I hope that our work opens possibilities for new applications and will be helpful to others in their own pursuits.

Appendix A

Computing messages $m^t(x)$ in LG-SSIM

Note that

$$\begin{aligned}
&\prod_{j=1}^{m} \mathcal{N}(X, \mu_j, \Sigma_j) \propto \prod_{j=1}^{m} e^{-\frac{1}{2}(X-\mu_j)^T \Sigma_j^{-1}(X-\mu_j)} \propto \prod_{j=1}^{m} e^{-\frac{1}{2}(X^T\Sigma_j^{-1}X - X^T\Sigma_j^{-1}\mu_j - \mu_j^T\Sigma_j^{-1}X)} \\
&= e^{-\frac{1}{2}(X^T \sum_{j=1}^{m} \Sigma_j^{-1} X - X^T \sum_{j=1}^{m} \Sigma_j^{-1}\mu_j - \sum_{j=1}^{m} \mu_j^T \Sigma_j^{-1} X)} \\
&= e^{-\frac{1}{2}(X^T \Sigma^{*-1} X - X^T \Sigma^{*-1}\Sigma^* \sum_{j=1}^{m} \Sigma_j^{-1}\mu_j - \sum_{j=1}^{m} \mu_j^T \Sigma_j^{-1}\Sigma^*\Sigma^{*-1} X)} \\
&\propto \mathcal{N}(X, \mu^*, \Sigma^*),
\end{aligned} \tag{A.1}$$

where

$$\Sigma^{*-1} = \sum_{j=1}^{m} \Sigma_j^{-1}, \quad \mu^* = \Sigma^* \sum_{j=1}^{m} \Sigma_j^{-1}\mu_j. \tag{A.2}$$

Messages $m^t(X_t)$, $t = 0, \ldots, T-1$, are computed in the following way:

$$m^t(X_t) = \int_{X_{t+1}} P(X_{t+1}|X_t, E_{Z_{t+1}}, \Theta_{Z_{t+1}}) P(Y_{t+1}|X_{t+1})\, m^{t+1}(X_{t+1})\, dX_{t+1}$$

$$= \int_{X_{t+1}} \mathcal{N}(X_{t+1}; A_{Z_{t+1}} X_t, \Sigma_{Z_{t+1}}) \underbrace{\mathcal{N}(Y_{t+1}; X_{t+1}, \Sigma_e)\mathcal{N}(X_{t+1}; \mu^m_{t+1}, \Sigma^m_{t+1})}_{\propto \mathcal{N}(X_{t+1}; \mu^o_t, \Sigma^o_t) \text{ by A.1}} dX_{t+1}$$

$$\propto \int_{X_{t+1}} e^{-\frac{1}{2}\left(X_{t+1} - A_{Z_{t+1}} X_t\right)^T \Sigma^{-1}_{Z_{t+1}} \left(X_{t+1} - A_{Z_{t+1}} X_t\right)} e^{-\frac{1}{2}\left(X_{t+1} - \mu^o_t\right)^T \Sigma^{o-1}_t \left(X_{t+1} - \mu^o_t\right)} dX_{t+1}$$

$$= \int_{X_{t+1}} e^{-\frac{1}{2}\Bigg[X^T_{t+1} \underbrace{\left(\Sigma^{-1}_{Z_{t+1}} + \Sigma^{o-1}_t\right)}_{\Sigma^{*-1}_t} X_{t+1} - X^T_{t+1} \Sigma^{*-1}_t \underbrace{\Sigma^*_t \left(\Sigma^{-1}_{Z_{t+1}} A_{Z_{t+1}} X_t + \Sigma^{o-1}_t \mu^o_t\right)}_{\mu^*_t} - \underbrace{\left(\left(A_{Z_{t+1}} X_t\right)^T \Sigma^{-1}_{Z_{t+1}} + \mu^{oT}_t \Sigma^{o-1}_t\right) \Sigma^*_t}_{\mu^{*T}_t} \Sigma^{*-1}_t X$$

$$\cdot e^{-\frac{1}{2}\left(A_{Z_{t+1}} X_t\right)^T \Sigma^{-1}_{Z_{t+1}} A_{Z_{t+1}} X_t} dX_{t+1}$$

$$= \int_{X_{t+1}} e^{-\frac{1}{2}\left[(X_{t+1} - \mu^*_t)^T \Sigma^{*-1}_t (X_{t+1} - \mu^*_t) - \mu^{*T}_t \Sigma^{*-1}_t \mu^*_t + \left(A_{Z_{t+1}} X_t\right)^T \Sigma^{-1}_{Z_{t+1}} A_{Z_{t+1}} X_t\right]} dX_{t+1}$$

$$\propto e^{-\frac{1}{2}\left[\left(A_{Z_{t+1}} X_t\right)^T \Sigma^{-1}_{Z_{t+1}} A_{Z_{t+1}} X_t\right) - \left(\left(A_{Z_{t+1}} X_t\right)^T \Sigma^{-1}_{Z_{t+1}} + \mu^{oT}_t \Sigma^{o-1}_t\right) \Sigma^*_t \Sigma^{*-1}_t \Sigma^*_t \left(\Sigma^{-1}_{Z_{t+1}} A_{Z_{t+1}} X_t + \Sigma^{o-1}_t \mu^o_t\right]}$$

$$\propto e^{-\frac{1}{2}\left[X^T_t A^T_{Z_{t+1}} \Sigma^{-1}_{Z_{t+1}} A_{Z_{t+1}} X_t - X^T_t A^T_{Z_{t+1}} \Sigma^{-1}_{Z_{t+1}} \Sigma^*_t \Sigma^{-1}_{Z_{t+1}} A_{Z_{t+1}} X_t - X^T_t A^T_{Z_{t+1}} \Sigma^{-1}_{Z_{t+1}} \Sigma^*_t \Sigma^{o-1}_t \mu^o_t - \mu^{oT}_t \Sigma^{o-1}_t \Sigma^*_t \Sigma^{-1}_{Z_{t+1}} A_{Z_{t+1}} X_t\right]}$$

$$= e^{-\frac{1}{2}\Bigg[X^T_t \underbrace{A^T_{Z_{t+1}} \left(\Sigma^{-1}_{Z_{t+1}} - \Sigma^{-1}_{Z_{t+1}} \Sigma^*_t \Sigma^{-1}_{Z_{t+1}}\right) A_{Z_{t+1}}}_{(\Sigma^m_t)^{-1}} X_t - X^T_t (\Sigma^m_t)^{-1} \underbrace{\Sigma^m_t A^T_{Z_{t+1}} \Sigma^{-1}_{Z_{t+1}} \Sigma^*_t \Sigma^{o-1}_t \mu^o_t}_{\mu^m_t} - \underbrace{\mu^{oT}_t \Sigma^{o-1}_t \Sigma^*_t \Sigma^{-1}_{Z_{t+1}} A_{Z_{t+1}} \Sigma^m_t}_{(\mu^m_t)^T} (\Sigma^m_t)^{-1}$$

$$\propto \mathcal{N}(X_t; \mu^m_t, \Sigma^m_t), \tag{A.3}$$

with

$$\begin{aligned}
(\Sigma^m_t)^{-1} &= A^T_{Z_{t+1}} (\Sigma^{-1}_{Z_{t+1}} - \Sigma^{-1}_{Z_{t+1}} \Sigma^*_t \Sigma^{-1}_{Z_{t+1}}) A_{Z_{t+1}} \\
\mu^m_t &= \Sigma^m_t A^T_{Z_{t+1}} \Sigma^{-1}_{Z_{t+1}} \Sigma^*_t \Sigma^{o-1}_t \mu^o_t \\
\Sigma^{*-1}_t &= \Sigma^{-1}_{Z_{t+1}} + \Sigma^{o-1}_t,
\end{aligned} \tag{A.4}$$

and, from Equations A.1 and A.2,

$$\begin{aligned}
\Sigma^{o-1}_t &= \Sigma^{-1}_e + {\Sigma^m_{t+1}}^{-1} \\
\mu^o_t &= \Sigma^o_t (\Sigma^{-1}_e Y_{t+1} + {\Sigma^m_{t+1}}^{-1} \mu^m_{t+1}).
\end{aligned} \tag{A.5}$$

Bibliography

[1] Hirotugu Akaike. A new look at the statistical model identification. *Automatic Control, IEEE Transactions on*, 19(6):716–723, 1974.

[2] M. Sanjeev Arulampalam, Simon Maskell, and Neil Gordon. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50:174–188, 2002.

[3] Michèle Basseville and Igor V. Nikiforov. *Detection of abrupt changes: theory and application*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993. ISBN 0-13-126780-9.

[4] Jim L Beck and Lambros S Katafygiotis. Updating models and their uncertainties. i: Bayesian statistical framework. *Journal of Engineering Mechanics*, 124(4):455–461, 1998.

[5] HS Bhat and N Kumar. On the derivation of the bayesian information criterion. 2010.

[6] James MW Brownjohn. Structural health monitoring of civil infrastructure. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 365(1851):589–622, 2007.

[7] Wray Buntine. Theory refinement on bayesian networks. In *Proceedings of the Seventh Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-91)*, pages 52–60, San Mateo, CA, 1991. Morgan Kaufmann.

[8] George Casella. An introduction to empiricaayes data analysis. *The American Statistician*, 39(2):83–87, 1985.

[9] Mehmet Çelebi, Nafi Toksöz, and Oral Büyüköztürk. Rocking behavior of an instrumented unique building on the mit campus identified from ambient shaking data. *Earthquake Spectra*, 30(2):705–720, 2014.

[10] David M. Chickering. Learning Bayesian networks is NP-Complete. In D. Fisher and H. Lenz, editors, *Learning from Data: Artificial Intelligence and Statistics V*, pages 121–130. Springer-Verlag, 1996.

[11] Gregory F. Cooper and Tom Dietterich. A bayesian method for the induction of probabilistic networks from data. In *Machine Learning*, pages 309–347, 1992.

[12] Luc Devroye. *Non-Uniform Random Variate Generation.* (originally published with Springer-Verlag), 1986.

[13] Zoran Dzunic and John Fisher III. Bayesian switching interaction analysis under uncertainty. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, pages 220–228, 2014.

[14] Zoran Dzunic, Justin G Chen, Hossein Mobahi, Oral Buyukozturk, and John W Fisher III. A bayesian state-space approach for damage detection and classification. In *Dynamics of Civil Structures, Volume 2*, pages 171–183. Springer, 2015.

[15] Eric B Flynn and Michael D Todd. A bayesian approach to optimal sensor placement for structural health monitoring with application to active sensing. *Mechanical Systems and Signal Processing*, 24(4):891–903, 2010.

[16] E.B. Fox, E.B. Sudderth, M.I. Jordan, and A.S. Willsky. Bayesian Nonparametric Inference of Switching Dynamic Linear Models. *IEEE Transactions on Signal Processing*, 59(4), 2011.

[17] Emily B. Fox, Erik B. Sudderth, Michael I. Jordan, and Alan S. Willsky. Nonparametric bayesian learning of switching linear dynamical systems. In Daphne Koller, Dale Schuurmans, Yoshua Bengio, and Lon Bottou, editors, *NIPS*, pages 457–464. Curran Associates, Inc., 2008.

[18] Emily B Fox, Erik B Sudderth, Michael I Jordan, and Alan S Willsky. An hdp-hmm for systems with state persistence. In *Proceedings of the 25th international conference on Machine learning*, pages 312–319. ACM, 2008.

[19] N. Friedman and D. Koller. Being Bayesian about Bayesian network structure: A Bayesian approach to structure discovery in Bayesian networks. *Machine Learning*, 50(1–2):95–125, 2003. Full version of UAI 2000 paper.

[20] Nir Friedman, Kevin P. Murphy, and Stuart J. Russell. Learning the structure of dynamic probabilistic networks. In *UAI*, pages 139–147, 1998.

[21] Sylvia Frhwirth-Schnatter. Fully bayesian analysis of switching gaussian state space models. *Annals of the Institute of Statistical Mathematics*, 53(1):31–49, 2001.

[22] Zoubin Ghahramani and Geoffrey E. Hinton. Switching state-space models. Technical report, Kings College Road, Toronto M5S 3H5, 1996.

[23] Clive WJ Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: Journal of the Econometric Society*, pages 424–438, 1969.

[24] Clive WJ Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: Journal of the Econometric Society*, pages 424–438, 1969.

[25] Marco Grzegorczyk and Dirk Husmeier. Improving the structure mcmc sampler for bayesian networks by introducing a new edge reversal move. *Mach. Learn.*, 71 (2-3):265–305, June 2008.

[26] Joanna D. Haigh. The sun and the earth's climate. *Living Reviews in Solar Physics*, 4(2), 2007.

[27] David Heckerman, Dan Geiger, and David M. Chickering. Learning bayesian networks: The combination of knowledge and statistical data. In *MACHINE LEARNING*, pages 197–243, 1995.

[28] Clifford M Hurvich and Chih-Ling Tsai. Regression and time series model selection in small samples. *Biometrika*, 76(2):297–307, 1989.

[29] Intergovernmental. *Climate Change 2007 - The Physical Science Basis: Working Group I Contribution to the Fourth Assessment Report of the IPCC.* Cambridge University Press, September 2007.

[30] Alan T James. Distributions of matrix variates and latent roots derived from normal samples. *The Annals of Mathematical Statistics*, pages 475–501, 1964.

[31] Huijing Jiang, Aurelie C. Lozano, and Fei Liu. A bayesian markov-switching model for sparse dynamic network estimation. In *SDM*, pages 506–515. SIAM / Omnipress, 2012.

[32] Mladen Kolar, Le Song, Amr Ahmed, Eric P Xing, et al. Estimating time-varying networks. *The Annals of Applied Statistics*, 4(1):94–123, 2010.

[33] François Le Gall. Powers of tensors and fast matrix multiplication. In *Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation*, ISSAC '14, pages 296–303, New York, NY, USA, 2014. ACM.

[34] Judith Lean and David Rind. Climate Forcing by Changing Solar Radiation. *Journal of Climate*, 11(12):3069–3094, December 1998.

[35] Sophie Lebre, Jennifer Becq, Frederic Devaux, Michael Stumpf, and Gaelle Lelandais. Statistical inference of the time-varying structure of gene-regulation networks. *BMC Systems Biology*, 4(1):130, 2010.

[36] David Madigan, Jeremy York, and Denis Allard. Bayesian Graphical Models for Discrete Data. *International Statistical Review / Revue Internationale de Statistique*, 63(2):215–232, 1995.

[37] Marina Meilă and Tommi Jaakkola. Tractable bayesian learning of tree belief networks. In *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, UAI'00, pages 380–388, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[38] Charles E. Metz. Basic principles of ROC analysis. *Seminars in Nuclear Medicine*, 8(4):283 – 298, 1978.

[39] Teppo Niinimaki, Pekka Parviainen, and Mikko Koivisto. Partial order mcmc for structure discovery in bayesian networks. In *Proceedings of the Twenty-Seventh Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-11)*, pages 557–564, Corvallis, Oregon, 2011. AUAI Press.

[40] Earth System Research Laboratory of the National Oceanic and Atmospheric Administration (NOAA). Climate indices: Monthly atmospheric and ocean time series. `http://www.esrl.noaa.gov/psd/data/climateindices/list/`. Accessed: 2013-10-21.

[41] Sang Min Oh, James M Rehg, Tucker Balch, and Frank Dellaert. Learning and inference in parametric switching linear dynamic systems. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1161–1168. IEEE, 2005.

[42] Vladimir Pavlovic, James M Rehg, Tat-Jen Cham, and Kevin P Murphy. A dynamic bayesian network approach to figure tracking using learned dynamic models. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 94–101. IEEE, 1999.

[43] Vladimir Pavlovic, James M. Rehg, and John MacCormick. Learning Switching Linear Models of Human Motion. In *Neural Information Processing Systems*, 2000.

[44] Judea Pearl. *Causality*. Cambridge university press, 2009.

[45] Aleksey S Polunchenko and Alexander G Tartakovsky. State-of-the-art in sequential change-point detection. *Methodology and Computing in Applied Probability*, 14 (3):649–684, 2012.

[46] Joshua W. Robinson and Alexander J. Hartemink. Learning non-stationary dynamic bayesian networks. *J. Mach. Learn. Res.*, 11:3647–3680, December 2010.

[47] R. W. Robinson. Counting labeled acyclic digraphs. *New Directions in the Theory of Graphs, Academic Press, New York*, 1973.

[48] Gideon Schwarz et al. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978.

[49] M. R. Siracusa. *Dynamic Dependence Analysis: Modeling and Inference of Changing Dependence Among Multiple Time-Series.* PhD thesis, MIT, Department of Electrical Engineering and Computer Science, 2009.

[50] Michael R. Siracusa and John W. Fisher III. Tractable bayesian inference of time-series dependence structure. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, 2009.

[51] Hoon Sohn, Charles R Farrar, Francois M Hemez, Devin D Shunk, Daniel W Stinemates, Brett R Nadler, and Jerry J Czarnecki. *A review of structural health monitoring literature: 1996-2001.* Los Alamos National Laboratory Los Alamos, NM, 2004.

[52] Le Song, Mladen Kolar, and Eric Xing. Time-Varying Dynamic Bayesian Networks. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 1732–1740. 2009.

[53] P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search.* MIT press, 2nd edition, 2000.

[54] Volker Strassen. Gaussian elimination is not optimal. *Numerische Mathematik*, 13 (4):354–356, 1969.

[55] Yee Whye Teh, Michaèl I Jordan, Matthew J Beal, and David M Blei. Hierarchical dirichlet processes. *Journal of the american statistical association*, 101(476), 2006.

[56] Michael W Vanik, JL Beck, and SK2000 Au. Bayesian probabilistic approach to structural health monitoring. *Journal of Engineering Mechanics*, 126(7):738–745, 2000.

[57] Thomas Verma and Judea Pearl. A theory of inferred causation. In *Second Internation Coference on the Principles of Knowledge Representation and Reasoning*, Cambridge, Massachusetts, April 1991.

[58] B. Walsh. Markov chain monte carlo and gibbs sampling, 2004.

[59] Xiang Xuan and Kevin Murphy. Modeling changing dependency structure in multivariate time series. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 1055–1062, New York, NY, USA, 2007. ACM.

[60] K. Yamasaki, A. Gozolchiani, and S. Havlin. Climate Networks around the Globe are Significantly Affected by El Ni[n-tilde]o. *Physical Review Letters*, 100(22), 2008.