Unraveling temporal processes using probabilistic graphical models

de Paula Bueno, M.L.

Unraveling Temporal Processes using Probabilistic Graphical Models

MARCOS LUIZ DE PAULA BUENO

Cover design: Matheus de Paula Bueno
Cover background image: Davide Guglielmo
Printed by Gildeprint
ISBN: 9789464020519

Nederlandse Organisatie voor Wetenschappelijk Onderzoek

SIKS Dissertation Series No. 2020-02.
The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

This thesis was supported by the Netherlands Organization for Scientific Research (NWO) as part of the "Careful" project (62001863), and by project "NORTE-01-0145-FEDER-000016" (NanoSTIMA) financed by the North Portugal Regional Operational Programme (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement, and through the European Regional Development Fund (ERDF).

Unraveling Temporal Processes using Probabilistic Graphical Models

Proefschrift

ter verkrijging van
de graad van Doctor aan de Universiteit Leiden,
op gezag van Rector Magnificus prof.mr. C.J.J.M. Stolker,
volgens besluit van het College voor Promoties
te verdedigen op dinsdag 11 februari 2020
klokke 15.00 uur

door

MARCOS LUIZ DE PAULA BUENO

geboren te Catalão, Brazilië
in 1984

Promotor:	Prof. dr. P.J.F. Lucas
Copromotor:	Dr. A.J. Hommersom (Open Universiteit)
Promotiecommissie:	Prof. dr. T.H.W. Bäck (secretaris)
	Prof. dr. A. Plaat (voorzitter)
	Prof. dr. M. Druzdzel (University of Pittsburgh, USA)
	Dr. S. Renooij (Universiteit Utrecht)

CONTENTS

1

INTRODUCTION

1.1 THE RELEVANCE OF TEMPORAL INFORMATION

The comprehension of real-world phenomena is often challenging, as their characterization might depend on some notion of time. A resulting lack of insight may stem from the fact that a single snapshot of a temporal process reveals only a part of its behavior, which may be insufficient for its complete understanding. This is the case, for example, when one contrasts a single instance of the observation of symptoms of a patient with a chronic disease against the longitudinal view of multiple instances of observations of symptoms: whereas the latter will offer a temporal view of the underlying disease process, the former will not shed any light upon disease dynamics.

In many everyday tasks, such as walking, cooking, sleeping and so on a role of temporal information can be identified. In professional fields, such as for example in psychiatry, the efficacy of pharmacological interventions in mental disorders only can be properly studied when the research is supported by collecting temporal data [3]. Such temporal information will tell for example how long it takes before a treatment becomes effective and how long and how often a patient should take a particular drug. Also in many other diseases, in particular those with a chronic duration, temporal information is of paramount importance to gain insight into speed of progress or recovery.

Thus, given the importance of information about time in everyday and professional life, when one wishes to mathematically *model* processes of human artifacts, for example cyberphysical systems, or processes in the life sciences, usually time will be one of the parameters that need to be taken into account. It is not surprizing that predictions about the future are often more accurate when taking into account the history than when not relying on such information [63]. Of course, reasoning with time not only is concerned with predicting the future given the past, but can go in any other direction: from the present going back in time to understand the past, or from assumptions in the future going back in time to understand which past conditions are needed to make a particular future feasible. Whether these kinds of temporal reasoning are possible is determined by the nature and capabilities of the mathematical models and reasoning methods employed.

In this thesis, temporal processes are modeled as stochastic processes. Such processes typically involve one or more random variables that can be repeatedly observed. More importantly to their characterization, however, is that past observations have influence on future observations, which assigns to a temporal process a sequential nature. In other words, it is not just a matter of merely observing variables at different moments, or making repeated measurements of variables. On the other hand, by considering the sequential nature of such processes, additional challenges are introduced due to the increased modeling complexities that come along.

1.2 PROBABILISTIC GRAPHICAL MODELS

An innate property of temporal processes is change, which renders them a stochastic nature. This makes probability theory a suitable tool for modeling such processes. Probabilistic models naturally take into account uncertainty, and can be used for multiple purposes: to predict the behavior of process variables in the future, discover associations between variables (e.g. which variables have more or less influence on a certain variable), and to pinpoint causes that could explain abnormal behavior.

Deriving models from data is a reality nowadays. This is because not only data storage technology has advanced (e.g. hardware capacity), but also more data is currently being generated, by means of sensor devices, content posted on the Internet, hospitals, health care services, etc. Obtaining statistical models from data which are expressive enough and can provide answers in reasonable running time is, however, not trivial. One major reason is that the process variables might interact in a very large number of ways. Without prior knowledge on the problem at hand, there is typically no obvious way as to how to reduce the space of models that might be of interest. In the past, researchers often relied on overly simplistic models (see e.g. [70]) to make model building feasible.

One solution to the parsimony problem faced by researchers is found with the adoption of probabilistic graphical models (PGMs, for short) [104, 136]. PGMs combine probabilities with graph theory for providing a graphical representation of probability distributions. With the representation of PGMs it becomes much easier to represent statistical properties suitable for the problem at hand. Well-known PGMs include Bayesian networks, hidden Markov models, Markov random fields, among others. PGMs allow for a move from probability distributions, which are rich in detail, to graphs that abstract away from such details by encoding independence relationships. This occurs by means of a *qualitative* semantics entailed by the graphical structure of PGMs.

The qualitative information encoded in a PGM is appealing for domain experts, who can read off relationships from the graph, such as whether a variable A becomes irrelevant for the prediction of B when C is known. The graphical representation allows for answering such queries often in a computationally efficient way. This represents an alternative way to identify independence properties by not relying on calculating numerical probabilities that can be computationally

demanding. PGMs also have a *quantitative* semantics by assigning numerical parameters to nodes in the graph, which allows for computing probability queries with full detail.

When it comes to model building, a number of advantages result from using graphs to represent distribution properties. At the domain-expert level, it is possible to specify the desired level of restrictions on the way variables can interact probabilistically. For example, one might say that the variables should be independent (leading to an empty graph), or that interactions should follow a tree-like pattern, or even specify detailed interactions. The language of graphs is intuitive enough to allow one to easily specify such patterns. Independence relationships between variables can also be derived in a completely algorithmic manner without any expert knowledge, as methods have been developed for learning the graphical structure of PGMs from data [163, 169]. A hybrid approach is also possible by combining expert knowledge with data-driven learning.

1.3 MODELING SEQUENTIAL BEHAVIORS

Dynamic Bayesian networks (DBNs, for short) [68, 104, 124] are well-known PGMs used for representing temporal processes. The process that a DBN models is assumed to be a first-order process (or memoryless), which means that the future state of the process depends only the present state. The process is also assumed to be time homogeneous, which means that probabilities for transitioning from time t to time $t+1$ are the same for every $t \geq 0$. As a result, DBNs offer a parsimonious representation of temporal processes by requiring the specification of typically a small number of probability parameters. DBNs can also be seen as extensions of discrete Markov chains to multiple variables. Example 1.1 discusses the dynamics of a mental disorder treatment based on DBNs.

Example 1.1. *Suppose we want to model the symptom dynamics of patients with psychotic depression [147], which is a depressive disorder with psychotic features. On a regular basis, two psychotic features (Delusions and Hallucinations) and Depression are measured for each patient. Due to the DBN assumptions aforementioned, the structure of a DBN for this problem is given by:*

- *A graph over the variables {Delusions, Hallucinations, Depression} indicating the symptom interactions at $t = 0$.*
- *A graph over $\{Delusions^{(t)}, Hallucinations^{(t)}, Depression^{(t)}, Delusions^{(t+1)}, Hallucinations^{(t+1)}, Depression^{(t+1)}\}$ indicating the transitioning interaction of symptoms at two any time points, where $t \geq 0$.*

The transition structure of a DBN for patient dynamics is shown in Figure 1.1. Although Figure 1.1 shows only one transition, this model would normally be unrolled for any discrete time horizon $\{0, 1, \ldots\}$. The transition structure and numerical parameters are the same for every transition due to the homogeneity assumption.

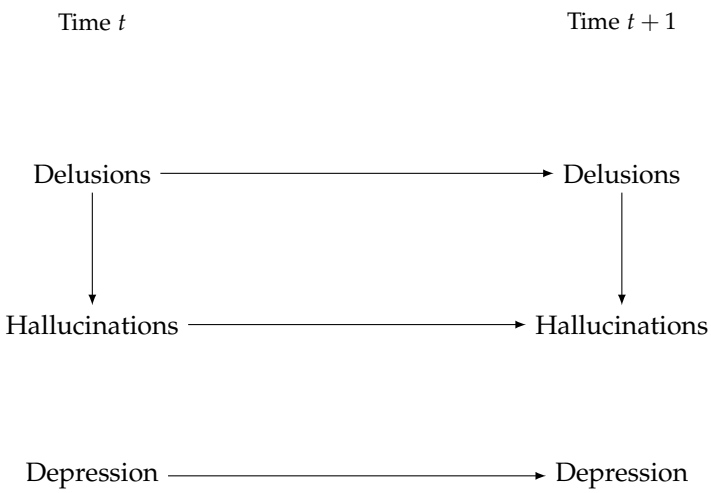


Figure 1.1: Transition structure of a dynamic Bayesian network that represents symptom interaction of psychotic depression patients.

In Figure 1.1, an arc between two variables indicates that these variables may be statistically dependent. On the other hand, two variables that are indirectly linked in the graph can still be statistically dependent, however this depends on the configuration of other variables in between them. When two variables are not linked directly nor indirectly, they are statistically independent [136].

Each arc of a DBN can refer to an *instantaneous* interaction, i.e. an interaction within the same time point, or a *temporal* interaction, i.e. an interaction that occurs at different time points [104]. The temporal arcs should satisfy the natural temporal order, i.e., there must be no arcs with direction from future to past. In Example 1.1, the instantaneous arc from Delusions to Hallucinations indicates that at any point where measurements were made, delusions and hallucinations are statistically correlated. On the other hand, the temporal arcs for current depression to future depression means that the depression score of a patient at a certain week has influence on the patient's depression the week after (as one would expect in general).

What makes the model of Figure 1.1 a temporal model is the temporal arcs, because otherwise variables at a time point t_2 would be statistically independent of any variable at time $t_1 < t_2$. The instantaneous interactions are not strictly needed for the model to be temporal, but they are often used to represent more complex statistical relationships.

As the process is assumed to be time homogeneous, in the model of Example 1.1 not only the graphical structure is fixed over time, but also the numerical parameters that describe the transition probabilities. Just as the initial graph structure, the initial numerical parameters might also differ from the transition parameters.

1.4 ADDING MORE EXPRESSIVE POWER

One consequence of the compactness of models as DBNs is that they capture the *average* process behavior over time. This is because by having a transition model that is time invariant, the structure and numerical parameters of a DBN are the same for every time point. However, this might not always be desirable, because processes might change over time. By restraining ourselves to models that represent average behaviors, we might lose the opportunity to learn important insight about processes.

Process change might be captured by different ways. One would expect that real processes are in constant change. However, we would like to capture sensible process changes in our models, which would allow us to arrive at parsimonious explanations of the process at hand. We discuss 3 situations (or *challenges*) where it is desirable to represent processes in a more expressive way.

1.4.1 *Time-dependent representation*

One situation where we might be interested in more expressive models occurs when we wish to capture process change that manifests by varying variable interaction over time. For example, if the instantaneous and temporal interactions between Hallucinations and Depression are substantially different over time in Example 1.1, one would likely obtain a better understanding of patient evolution by modeling symptom interaction change in an explicit fashion.

In the case of models such as DBNs, process change could manifest as changes in the graphical structure, numerical parameters, or both. One challenge that arises in modeling is the identification of process change-points (or regime change) which take into account reasonable process change assumptions.

1.4.2 *Factor-dependent representation*

A different way of looking at process change is by identifying latent variables. Such factors can be seen as unmeasured quantities that are missing or difficult to be measured [178]. Latent variables can also be seen as categories or abstract concepts derived from observable data, such as intelligence and extraversion in the context of human behavior [17]. In the latter case, latent variables might act as a dimensionality reduction tool of observable data as it might be easier to understand the data in terms of the usually more compact latent representation.

In Example 1.1, latent factors that might be associated to process change could be medical treatment, environmental factors (age, gender, climate, etc.), genetic expression, etc. Some of these factors might actually happen to be measured (e.g. medication and dosage), which would allow for explicitly including them as observable variables. At other times we are interested in learning latent concepts from observable data, such as patient clusters in Example 1.1, which might help us understand patient dynamics by means of a succinct representation.

Hidden Markov models (HMMs, for short) [138, 141] are PGMs that use latent variables to model sequential processes. HMMs can capture very complex distributions by using a suitable latent state space [13]. The standard assumption in HMMs is that observable variables are independent given the latent state. However, representing complex observed behavior based on such assumption can require too many states [76]. This is undesirable for several reasons, e.g., computing cost and less interpretable models.

Extensions of HMMs that are able to represent more general variable interaction have been proposed [12, 76, 102]. One challenge that arises is a better understanding of how such HMMs compare in theory and in practice. Another relevant challenge is how to generate problem insight from more expressive HMMs which can help one understand the dynamics of processes (e.g. disease processes) in an effective way. This is relevant because the task of interpreting latent states is usually not straightforward.

1.4.3 *Subprocess representation*

A different viewpoint on process change concerns the identification of *subprocesses* which deviate considerably from the main model (or main process). We refer to 'main process' and 'subprocess' as processes associated to the model of the whole dataset and the model of a subset of the data respectively. In this situation, we would like to identify subsets of the data that are also representative enough, because it is easy to come up with very specific subprocesses made out of just a few data points (hence, not representative).

In Example 1.1, there could exist a subset of patients with certain psychotic symptoms that have a substantially slower response to treatment than the average patient response. We would like to identify such subsets (or subprocesses) in an automatic fashion. However, we cannot directly identify such subprocesses by models such as standard DBNs and HMMs.

One approach to identifying deviating subprocesses is the exceptional model mining (EMM, for short) [57, 110], whose goal is the discovery of exceptional models (in the sense of significantly different) associated to subsets of the data. However, EMM has been limited to either static data or univariate temporal data [112]. It would be desirable to extend the EMM framework for more general temporal data.

1.5 THESIS OUTLINE

This work addresses the discovery of structure from temporal data that can aid the comprehension of the underlying processes. For convenience, this work is divided into three parts as follows. In Chapters 3-5, we investigate the underlying structure of processes by means of models that use latent variables. In Chapter 6, we investigate how temporal processes can be better understood by identifying process changepoints or regime change. Finally, in Chapter 7 we investigate

how temporal data can be decomposed based on data subgroups that have a substantially different characterization in terms of a set of target variables compared to the distribution of those targets in the whole data. As a result, Chapter 7 provides a different characterization of process structure compared to those of other chapters.

In order to demonstrate the usefulness of the proposed methods, we use real-world data from several domains, including medical data (e.g. primary care and clinical trials), industrial processes and business processes. We also discuss problem insight that can be obtained by the application of the methods to such datasets. We summarize the content of each chapter in the following.

CHAPTER 2: *Preliminaries*, where notions on PGMs relevant for representing temporal processes are discussed.

CHAPTER 3: *Asymmetric hidden Markov models*, where we introduce the family of *asymmetric hidden Markov models* (HMM-As, for short) for representing local structure of distributions in the hidden Markov model framework. An algorithm for learning HMM-As from temporal data is proposed. HMM-As are empirically evaluated based on simulated data and real-world data from several domains. This chapter is based on the publications [25] and [23].

CHAPTER 4: *Predicting disease dynamics: a case study of psychotic depression*, in which a methodology is proposed for aiding the generation of medical hypotheses based on structured hidden Markov models learned from data. The methodology is used to uncover insight on the dynamics of different pharmacological therapies undertaken by psychotic depression patients. This chapter is based on the publications [27] and [28].

CHAPTER 5: *Understanding multimorbidity through clusters of hidden states*, where we analyze the problem of disease interaction and multimorbidity in terms of patterns of transitions between latent states. We consider a study case of patients with disorders related to atherosclerosis based on a large primary care data, and show that multiple patient characterization can be associated to cluster of states. This chapter is based on the publication [26].

CHAPTER 6: *Partitioned dynamic Bayesian networks*, in which we propose *partitioned dynamic Bayesian networks* (PDBNs, for short) for representing temporal processes by means of a collection of dynamic Bayesian networks. We propose a learning algorithm for PDBNs which adds process cut-offs in a parsimonious way. PDBNs are evaluated experimentally based on simulations and real data. This chapter is based on the publication [24].

CHAPTER 7: *Exceptional model mining using dynamic Bayesian networks*, where we investigate how observable data can be decomposed in a way different than that pursed in the previous chapters. We propose a method to identify subgroups of data that are exceptionally different than the total data based on the framework of subgroup discovery and exceptional model mining.

Subgroups are characterized by dynamic Bayesian networks. This chapter is based on the paper [22], which was submitted for publication.

CHAPTER 8: *Discussion*, in which the results achieved in this work are summarized and future directions to be pursued are discussed.

2

PRELIMINARIES

In this chapter, we fix the notation used throughout this work and present definitions on probabilistic graphical models that are relevant for the following chapters. We start off by covering the basics of Bayesian networks, then move to dynamic Bayesian networks and hidden Markov models, which extend the framework of Bayesian networks for handling temporal problems. Learning models from data is also discussed.

2.1 NOTATION

We first introduce the notation and a few conventions used throughout this work. In probability theory, *random variables* are typically denoted by upper case letters, such as X, while the *domain* of a random variable X is represented by $\mathrm{dom}(X)$, which represents the set of values that X takes on [52]. A *discrete random variable* is a random variable which has a finite or countably infinite domain, while a *continuous random variable* has as domain a subset of the real numbers. A random variable is associated to a *probability distribution*, which assigns a probability value to each value of its domain (for discrete variables) or to real intervals of its domain (for continuous variables).

The probability distribution of a discrete random variable X will be denoted by $P(X)$, and the probability of a certain value $x \in \mathrm{dom}(X)$ will be denoted by $P(X = x)$ or simply $P(x)$ when no confusion can arise. The probability distribution of a continuous random variable Y with probability density function $f(Y)$ is denoted by $p(Y)$, and the probability that Y takes values on a real interval $[y_1, y_2]$ is indicated as $p(y_1 \leq Y \leq y_2)$. A set of random variables will be denoted by a bold face letter, e.g., $\mathbf{X} = \{X_1, \ldots, X_n\}$. A probability distribution assigned to a single random variable as in $P(X)$ is called a *univariate* distribution, while the joint distribution assigned to set of variables $\{X_1, \ldots, X_n\}$ is called a *multivariate* distribution and is denoted by $P(X_1, \ldots, X_n)$ or $P(\mathbf{X})$.

In *temporal modeling*, each variable is often measured repeatedly, such that a variable X at time t will be referred to as $X^{(t)}$. This means that the domains of $X^{(t)}$, for all $t \geq 0$, are the same. For the discrete time points $\{t_1, \ldots, t_2\}$, where $t_2 \geq t_1 \geq 0$, the notation $X^{(t_1:t_2)}$ will be used to refer to the set of variables $\{X^{(t_1)}, \ldots, X^{(t_2)}\}$.

2.2 BAYESIAN NETWORKS

2.2.1 *Origin*

A Bayesian network (BN, for short) is a graphical model of a multivariate probability distribution with independence constraints. Bayesian networks date back to the 1980s [97, 104, 136]; their goal was to overcome the limitations of rule-based expert systems from the previous decade that incorporated uncertainty in the form of numbers that had some resemblance to probabilities [115]. One important limitation of such AI systems was the need for representing an unrealistic number of probabilities to perform probabilistic inference, a problem which was dealt with by making many simplifying assumptions. While this led to a substantial reduction in the needed number of probabilistic parameters, it also gave rise to poor performance in solving real-world problems, for example in medical diagnosis [70]. By marrying probability theory with graph theory, Bayesian networks allowed to provide the right balance in the number of probabilistic parameters needed to realistically represent the problem domain at hand.

A Bayesian network is a two-fold representation, as it encodes both qualitative and quantitative information about probability distributions. The qualitative side of a Bayesian network is given by a graph, whose semantics is associated to statistical independence statements. The quantitative side regards numerical probabilities, which are specified following the structure of the graph. As a result, BNs provide a compact, yet expressive way of representing probability distributions.

2.2.2 *Representation*

To define Bayesian networks over a set of random variables of interest, a few definitions are introduced first. A *graph* $\mathcal{G}$ is a pair $\mathcal{G} = (\mathbf{V}, \mathbf{A})$, where $\mathbf{V}$ is a set of objects $i \in \{1, \dots, n\}$, $n = |\mathbf{V}|$, called *nodes*, and $\mathbf{A} \subseteq \mathbf{V} \times \mathbf{V}$ is a set of node pairs called *edges*. If $\mathcal{G}$ is a *directed graph*, then each edge of A is an ordered pair (i, j), also represented by $i \to j$, such that $(j, i) \notin \mathbf{A}$. The edges are then called *directed edges* or *arcs*. If $i \to j \in \mathbf{A}$ is an arc, then i is called the *parent* of node j, and j is called the *child* of node i. If there is a directed path from node i to node j, then i is called the *ancestor* of j, whereas j is called the *descendant* of i.

If $\mathcal{G}$ is an *undirected graph*, then its edges are unordered pairs, i.e., if $(i, j) \in \mathbf{A}$ then also $(j, i) \in \mathbf{A}$, simply represented as a set $\{i, j\}$. A *directed acyclic graph* (DAG, for short) is a directed graph with no cycles, i.e., there is no sequence of arcs of the form $i \to j \to \cdots \to i$ (first and last node in the sequence are the same). As usual, each node i in the DAG with $V = \{1, \dots, n\}$ will be associated in a one-to-one way to a random variable X_i from the set of variables $X_1, \dots, X_n$ for the convenience of defining a Bayesian network. In the following, we shall refer to nodes and variables interchangeably and use X_i to refer to both the node and the variable.

One way to define Bayesian networks is from the notion of factorizing a joint probability distribution according to the structure of a graph as follows.

Definition 2.1 (Factorization). *Let $\mathcal{G} = (\mathbf{V}, \mathbf{A})$ be a directed acyclic graph with nodes $\mathbf{V} = \{X_1, \ldots, X_n\}$. A joint probability distribution P over the same variables factorizes according to $\mathcal{G}$ if P can be written as:*

$$P(\mathbf{X}) = P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i \mid \pi(X_i)) \tag{2.1}$$

where $\pi(X_i)$ refers to the parents of the node X_i in $\mathcal{G}$, and each factor $P(X_i \mid \pi(X_i))$ is called a conditional probability table (CPT, for short).

Definition 2.2 (Bayesian network). *A Bayesian network is a pair $\mathcal{B} = (\mathcal{G}, P)$, where P is a joint probability distribution that factorizes according to a directed acyclic graph $\mathcal{G}$.*

The joint probability distribution P associated with a Bayesian network $\mathcal{G}$ encodes conditional independences, if it holds for three mutually disjoint sets of variables $\mathbf{U}, \mathbf{W}, \mathbf{Z} \subseteq \mathbf{X}$ that if $P(\mathbf{U} \mid \mathbf{W}, \mathbf{Z}) = P(\mathbf{U} \mid \mathbf{Z})$ for any set of values of the variables in $\mathbf{U}, \mathbf{W}, \mathbf{Z}$. It is said that the variables $\mathbf{U}$ and $\mathbf{W}$ are *conditionally independent* (under P) given $\mathbf{Z}$, written as $\mathbf{U} \perp\!\!\!\perp_P \mathbf{W} \mid \mathbf{Z}$. The set of all conditional independent triplets associated to a joint probability distribution P is sometimes defined as $I(P) = \{(\mathbf{U}, \mathbf{W}, \mathbf{Z}) \mid \mathbf{U} \perp\!\!\!\perp_P \mathbf{W} \mid \mathbf{Z}\}$.

The Bayesian network graph encodes independence relationships, which can be read off by means of a graphical property called *d-separation* (directed separation) [136]. The notion of d-separation defines potential probabilistic influence between variables based on the structure of the BN graph. This can be described by means of the notion of *active trail* [104]. A sequence of nodes $\sigma = X_1, \ldots, X_m$ in the graph $\mathcal{G}$ is a *trail* if either $X_i \to X_{i+1}$ or $X_i \leftarrow X_{i+1}$ is an arc in $\mathcal{G}$ on the trail σ, $i = 1, \ldots, m-1$, i.e., the direction of the arcs is ignored and only the fact that X_1 is connected by the trail to X_m is taken into account. Now the trail between X_1 and X_m is called *active* if for any Y on the trail σ, the connections of the neighboring nodes U and W have the following directions:

- $U \leftarrow Y \to W$ (divergent connection), $U \to Y \to W$ (serial connection), or $U \leftarrow Y \leftarrow W$ (serial connection), and Y has *not* been observed, or
- $U \to Y \leftarrow W$ (convergent connection or v-structure), whereas Y or any of its descendants have been observed.

If a trail is not active, it is called *inactive*.

Now consider the following three mutually disjoint sets of nodes $\mathbf{U}, \mathbf{W}, \mathbf{Z} \subseteq \mathbf{V}$. If all trails between any node in $\mathbf{U}$ and any node in $\mathbf{W}$ are inactive given (possibly empty) observations in $\mathbf{Z}$, it is said that the set of nodes $\mathbf{Z}$ *d-separates* the set of nodes $\mathbf{U}$ and $\mathbf{W}$, written as

$$\mathbf{U} \perp\!\!\!\perp^d_{\mathcal{G}} \mathbf{W} \mid \mathbf{Z}$$

For the graph $\mathcal{G}$ we can now collect all d-separation triplets:

$$I(\mathcal{G}) = \{(\mathbf{U}, \mathbf{W}, \mathbf{Z}) \mid \mathbf{U} \perp\!\!\!\perp^d_{\mathcal{G}} \mathbf{W} \mid \mathbf{Z}\}$$

The above definition can be used to provide a semantics of the BN graph in terms of independence statements that are entailed by the graph. For Bayesian networks $\mathcal{B} = (\mathcal{G}, P)$ where the distribution P factorizes according to the DAG $\mathcal{G}$, it holds that $I(\mathcal{G}) \subseteq I(P)$ [136]. This means that every independence that holds in the BN graph must hold in the distribution. This explains why the interpretation of d-separation as conditional independence is meaningful. However, the semantics makes also clear that the two independence relations $I(P)$ and $I(\mathcal{G})$ may not coincide.

Because of d-separation, the network structure of a Bayesian network can be seen as its *qualitative* part, while the *quantitative* part corresponds to the probabilities encoded in the CPTs. A Bayesian network example is provided in Example 2.1.

Example 2.1. *Assume we are interested in diagnosing lung cancer, as represented by the variable C with dom(C) = {no, yes}. Other variables of interest are smoking (S), gender (G), and age (A), where dom(S) = {no, yes}, dom(G) = {female, male} and dom(A) = {adult, elderly}. Figure 2.1 shows the graphical structure and the CPTs associated to this Bayesian network. Independence relationships can be deduced from the graphical structure, e.g., having knowledge about smoking will make age and gender irrelevant for the prediction of lung cancer.*

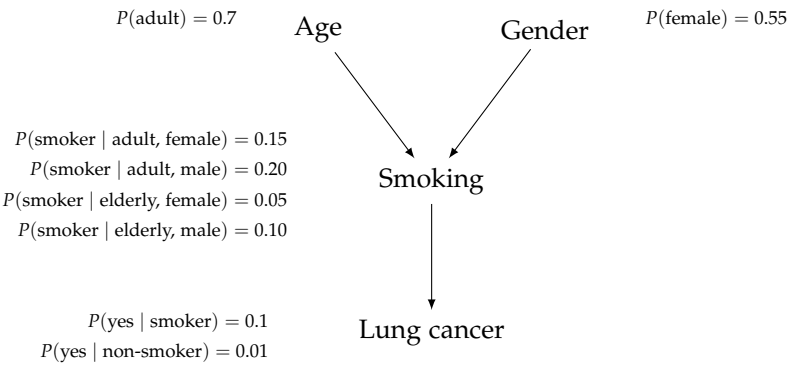


Figure 2.1: Bayesian network of the lung cancer example. Note that the structure and the CPTs in this example are fictional.

One advantage of the BN framework is that in the network structure one typically captures only the essential variable interactions, which often results in a substantial reductions in the needed number of probabilities to be specified. The BN of Example 2.1 requires $1 + 1 + 4 + 2 = 8$ independent parameters, while the explicit specification of the joint distribution of $P(A, G, S, C)$ would require $2^{(4)} - 1 = 15$ independent parameters. This advantage tends to be more significant if one deals with BNs with more variables, or variables with larger domains, for example.

The first step to using the BN representation is usually finding a suitable network structure for the problem at hand. One way to construct such a network structure is by manually defining the interactions among the involved variables that are supposed to hold in the domain, based on prior background knowledge and supported by domain experts. In that case, it is usually easier to think of interactions in terms of cause-effect relationships [47], although the semantics of a Bayesian network per se does not embody a causality notion. The network structure can also be obtained from data. In any case, once the network structure is obtained, the parameters of the network nodes need to be estimated, which can also be done manually [71] or algorithmically [47].

2.3 LEARNING BAYESIAN NETWORKS

In practical situations, prior knowledge about the problem at hand might not be available for handcrafting a Bayesian network for the domain, for example, if it is too expensive to be obtained, nonexistent, or is prone to be incorrect. This motivates the need for Bayesian network learning algorithms [44, 85, 169], whose goal is to automatically find a Bayesian network that suitably represents the distribution associated to the data. Bayesian network learning involves two steps: learning a network structure and learning numerical parameters. Handling each task also depends whether the data is complete or incomplete. In this section, we consider the case of complete data.

2.3.1 *Parameter learning*

In the parameter learning task, we assume the network structure is known. The goal is to estimate the CPTs of all the variables, i.e. the distributions $P(x_i \mid \pi(x_i))$ for every x_i. Let us denote by θ the set of parameters associated to the Bayesian network which are to be estimated, and let D be a set of data points $\mathbf{x}[1], \ldots, \mathbf{x}[m]$ of the form $\mathbf{x}[j] = \{x_1[j], \ldots, x_n[j]\}$, where $x_i[j]$ refers to the value of the variable X_i taken in the jth data point. Each X_i is assumed to follow a categorical distribution taking values on $\mathrm{dom}(X_i)$. We further assume that all data points of D are independent and identically distributed (i.e., i.i.d. samples). The likelihood function of the Bayesian network with structure $\mathcal{G}$ parameterized by θ given the data D corresponds to the probability of D under such model and is given by:

$$\mathcal{L}(\theta : D) = P(\mathbf{x}[1], \ldots, \mathbf{x}[m] : \theta) \tag{2.2}$$

$$= \prod_{j=1}^{m} P(\mathbf{x}[j] : \theta) \tag{2.3}$$

From the factorization of Bayesian networks we obtain:

$$\mathcal{L}(\theta\colon D) = \prod_{j=1}^{m}\prod_{i=1}^{n} P(x_i[j] \mid \pi(x_i)[j]\colon \theta) \tag{2.4}$$

$$= \prod_{i=1}^{n} P(x_i[1] \mid \pi(x_i)[1]\colon \theta) \ldots P(x_i[j] \mid \pi(x_i)[j]\colon \theta) \tag{2.5}$$

Equation 2.5 shows that the likelihood function can be decomposed into a product of independent terms, one for each node of the network structure. If we denote by θ_{irk} the parameter $P(X_i = x_k \mid \pi(X_i) = \mathbf{x}_r)$, then the likelihood function can be further expanded as follows:

$$\mathcal{L}(\theta\colon D) = \prod_{i=1}^{n}\prod_{x_k,\mathbf{x}_r} \theta_{irk}^{N_{irk}} \tag{2.6}$$

where N_{irk} is the number of times the configuration $(X_i = x_k, \pi(X_i) = \mathbf{x}_r)$ is seen in D. The goal now is to find the set of parameters θ that maximize this function, an approach known as *maximum likelihood estimation* (MLE, for short). The parameters that maximize the likelihood function are denoted by $\hat{\theta}$. It is usually easier to work with the logarithm of Equation 2.6, which is referred to as the log-likelihood of the data. It is possible to show [104] that the maximization of the log-likelihood leads to closed-form formulas for parameter learning as follows:

$$\hat{\theta}_{irk} = \frac{N_{irk}}{N_{ir}} \tag{2.7}$$

where N_{ir} is the number of times the configuration $(\pi(X_i) = \mathbf{x}_r)$ occurs in D. These quantities are known as *sufficient statistics*, and convey the idea that each parameter corresponds to the node's proportional counts with respect to its parents. Once the optimal parameters $\hat{\theta}$ are computed, the likelihood computed based on $\hat{\theta}$ is denoted by $\hat{\mathcal{L}}$.

2.3.2 *Structure learning*

When the network structure is unknown, one resorts to learning the network structure from data. The goal of structure learning is to recover the structure of the hypothetical joint probability distribution underlying the data [50]. With structure learning, one is able to discover the dependence structure of the domain, which can yield insight about qualitative influences that hold in the domain, both direct and indirect. Network structure learning is also important to make the parameter estimation in Bayesian networks feasible, although the structure should not be overly simplistic, otherwise relevant correlations might be missed.

The problem of structure learning can be formulated as an optimization problem [7, 42], also known as *score-based* approach, whose goal is to find the network structure $\hat{\mathcal{G}}$ that optimizes a scoring function:

$$\hat{\mathcal{G}} = \arg\max_{\mathcal{G}\in\boldsymbol{\mathcal{G}}} \text{Score}(\mathcal{G}, D) \tag{2.8}$$

where $\mathcal{G}$ is the space of network structures, i.e. the set of directed acyclic graphs with nodes $\{X_1, \ldots, X_n\}$. In general, finding optimal Bayesian networks has been shown intractable [42, 43]. Network structures can also be learned by means of the *constraint-based* approach [118, 136], which determines a network structure that is consistent with the independence relationships that hold on the data. In the case of constraint-based learning, the worst case requires an exponential number of tests [118]. In the remainder of this chapter, we consider the score-based approach and further elaborate on it.

Scoring functions for structure learning play a central role in this task and the literature offers a variety of them. One of the simplest score functions is the likelihood score [104], which indicates the probability of the data given the model and was defined in Equation 2.2. In the situation of unknown structure, the likelihood score seeks the model (i.e. graph and parameters) that maximizes the likelihood:

$$\max \mathcal{L}(\mathcal{G}, \theta_{\mathcal{G}} : D) = \max_{\mathcal{G}} \left[\max_{\theta_{\mathcal{G}} \in \Theta} \mathcal{L}(\mathcal{G}, \theta_{\mathcal{G}} : D) \right] \tag{2.9}$$

$$= \max_{\mathcal{G}} \left[\mathcal{L}(\mathcal{G}, \hat{\theta}_{\mathcal{G}} : D) \right] \tag{2.10}$$

where Θ is the space of CPTs with regard to the graph $\mathcal{G}$. Hence, in order to maximize the likelihood, one needs to find the structure $\hat{\mathcal{G}}$ that maximizes Equation 2.10, where each candidate structure has parameters fitted via MLE. However, as the goal with model learning is to capture the true distribution of the data, using the likelihood score typically has severe limitations as follows. By adding more arcs to the network, the likelihood score never decreases and instead tends to increase [104]. Hence, by completely fitting to the data, one is also fitting to the noise on the data, and the resulting network tends to be a fully connected graph. This usually leads to the problem of model overfitting, which means that the model does not generalize well (i.e. it performs poorly on new data).

One alternative score function is the Bayesian score, which adopts a Bayesian approach to modeling the structure and parameters that are to be estimated. In the Bayesian approach, one defines a structure prior $P(\mathcal{G})$ and a parameter prior $P(\theta_{\mathcal{G}} \mid \mathcal{G})$ for the possible ways a given structure can be parameterized. For a candidate graph $\mathcal{G}$, we can apply Bayes' rule to obtain:

$$P(\mathcal{G} \mid D) \propto P(D \mid \mathcal{G}) P(\mathcal{G}) \tag{2.11}$$

where the denominator $P(D)$ can be dropped because it is the same for all the graphs. The Bayesian score is then defined by taking the logarithm of the right-hand side of Equation 2.11:

$$\text{Score}_B(\mathcal{G} \mid D) = \log P(D \mid \mathcal{G}) + \log P(\mathcal{G}) \tag{2.12}$$

In the prior $P(\mathcal{G})$ one can model a prior distribution that might favor, e.g., sparser graphs. The term $P(D \mid \mathcal{G})$ is known as marginal likelihood as it can be written as:

$$P(D \mid \mathcal{G}) = \int_{\theta_{\mathcal{G}} \in \Theta} P(D \mid \theta_{\mathcal{G}}, \mathcal{G}) P(\theta_{\mathcal{G}} \mid \mathcal{G}) d\theta_{\mathcal{G}} \tag{2.13}$$

Intuitively, the marginal likelihood weights the likelihood of the data $P(D \mid \theta_{\mathcal{G}}, \mathcal{G})$ by different ways of selecting the parameters given the network $\mathcal{G}$. Hence, the marginal likelihood can be seen as an average of the likelihoods for the structure $\mathcal{G}$, as opposed to the maximum likelihood score, which looks only at the score that maximizes the term $P(D \mid \theta_{\mathcal{G}}, \mathcal{G})$. The Bayesian score tends to favor simpler structures if little data is available for learning [104], which provides a mechanism to combat overfitting. By using Dirichlet priors on all parameters of the network, it is possible to show [104] that an approximation of the Bayesian score results in the so-called Bayesian information criterion (BIC, for short), which is given as follows:

$$\text{BIC}(\mathcal{G} \mid D) = -2 \cdot \log \mathcal{L}(\hat{\theta}_{\mathcal{G}} : D) + K \cdot \log m \tag{2.14}$$

where K is the number of parameters of the network structure $\mathcal{G}$, and m is the size of the dataset.

The goal now is to find the structure $\mathcal{G}$ that minimizes the BIC score, where the term $K \cdot \log m$ in Equation 2.14 acts as a penalty term. Equation 2.14 suggests that the problem of structure learning can be seen as a model selection problem [182], where one wishes to find the network structure that balances goodness-of-fit (the likelihood term) and model size. The scoring function is then coupled to a search procedure, which is often a heuristic procedure such as tabu search [77], hill climbing, simulated annealing, among others [152], resulting in sub-optimal network structures obtained in feasible running time.

Although heuristic procedures are often used in BN structure learning, research has shown that optimal structure learning can be done efficiently in some situations [31, 158]. Some techniques are able to scale to problems with hundreds variables [50]. Research has also shown that it is possible to predict which algorithms would be more suitable for optimal learning of a given instance [116].

2.3.3 *Decomposable scores*

In structure learning, a key computational property is that of decomposability. A score is decomposable if it is defined locally per node [50]. This allows for the score of a candidate Bayesian network $\mathcal{B}$, also referred to as its *global score*, to be given as a sum of *local scores*, one for each variable:

$$\text{Score}(\mathcal{B}) = \sum_{X_i \in \mathbf{X}} \text{Score}(X_i, \pi_{\mathcal{B}}(X_i)) \tag{2.15}$$

where $\pi_{\mathcal{B}}(X_i)$ refers to the parents of X_i in $\mathcal{B}$. Decomposable scores allow for the efficient evaluation of small changes to the structure, such as arc removal and arc

addition, as such operations affect only the associated local scores. As a result, by exploiting this property, structure learning algorithms can scale reasonably well. Many scores commonly used in structure learning are decomposable, where the BIC is one such score [50].

2.4 DYNAMIC BAYESIAN NETWORKS

In this section we discuss extensions of Bayesian networks for modeling temporal processes by means of dynamic Bayesian networks (DBNs, for short). Learning DBNs from data is also discussed.

2.4.1 *Representation*

Dynamic Bayesian networks [68, 124] extend Bayesian networks for modeling temporal processes where uncertainty plays an important role. We restrain ourselves to dynamic systems in which all the variables of a set $\mathbf{X} = \{X_1, \ldots, X_n\}$ are measured together and repeatedly over time, which is represented by $\mathbf{X}^{(0)}, \mathbf{X}^{(1)}, \ldots$. Further, the time interval between two measurements $\mathbf{X}^{(t)}$ and $\mathbf{X}^{(t+1)}$, for any $t \geq 0$, is assumed fixed. This means that in such dynamic systems the sequential behavior of the involved variables is abstracted from the absolute time of their measurement.

In order to keep the model compact, a few additional assumptions about the process involved in the generation of $\mathbf{X}$ are often considered [104], which we describe as follows.

Definition 2.3 (Markovian dynamic system)**.** *A dynamic system over the variables* $\mathbf{X}$ *is first-order Markovian (or simply Markovian) if, for all* $t \geq 0$*,*

$$P(\mathbf{X}^{(t+1)} \mid \mathbf{X}^{(0:t)}) = P(\mathbf{X}^{(t+1)} \mid \mathbf{X}^{(t)}) \tag{2.16}$$

The Markovian assumption means that predicting the future state of the process depends only on its current state and not on previous states it has assumed. In this case, the process is also said to be *memoryless*. Another useful property is given as follows.

Definition 2.4 (Time-homogeneous dynamic system)**.** *A dynamic system over the variables* $\mathbf{X}$ *is time homogeneous (or time invariant) if* $P(\mathbf{X}^{(t+1)} \mid \mathbf{X}^{(t)})$ *is the same for every* $t \geq 0$*.*

Dynamic Bayesian networks provide a representation for Markovian time-homogeneous dynamic systems grounded on graphical models as defined next.

Definition 2.5 (Dynamic Bayesian network)**.** *A dynamic Bayesian network is a Markovian time-homogeneous system* $(\mathcal{B}_0, \mathcal{B}_{\rightarrow})$ *over* $\mathbf{X}$*, where:*

- $\mathcal{B}_0 = (\mathcal{G}_0, P_0)$ *is a Bayesian network over the variables* $\mathbf{X}^{(0)}$ *called initial network.*

- $\mathcal{B}_{\rightarrow} = (\mathcal{G}_{\rightarrow}, P_{\rightarrow})$ *is a Bayesian network over the variables* $\{\mathbf{X}^{(t)}, \mathbf{X}^{(t+1)}\}$ *called transition network. The variables of* $\mathbf{X}^{(t)}$ *have no parents in the transition network.*

The transition network can also be seen as a *conditional Bayesian network* [104], because it suffices to define the distribution $P(\mathbf{X}^{(t+1)} \mid \mathbf{X}^{(t)})$ for defining this network. Although DBNs can be defined as semi-infinite systems [68], in practice one reasons with a finite horizon $\{0, \ldots, T\}$. In this case, the DBN is unrolled so that a joint distribution over the process duration is specified as follows: the structure and parameters of all the nodes at time $t = 0$ come from the initial model, while the structure and parameters for any node $X_i^{(t)}$, where $t > 0$, come from the transition model.

From the previous definitions and assumptions, the joint distribution of a DBN over a time horizon $\{0, \ldots, T\}$ is as follows:

$$P(\mathbf{X}^{(0:T)}) = P(\mathbf{X}^{(0)}) \prod_{t=0}^{T-1} P(\mathbf{X}^{(t+1)} \mid \mathbf{X}^{(t)}) \tag{2.17}$$

$$= \prod_{i=1}^{n} P_0(X_i^{(0)} \mid \pi(X_i^{(0)})) \prod_{t=0}^{T-1} \prod_{i=1}^{n} P_{\rightarrow}(X_i^{(t+1)} \mid \pi(X_i^{(t+1)})) \tag{2.18}$$

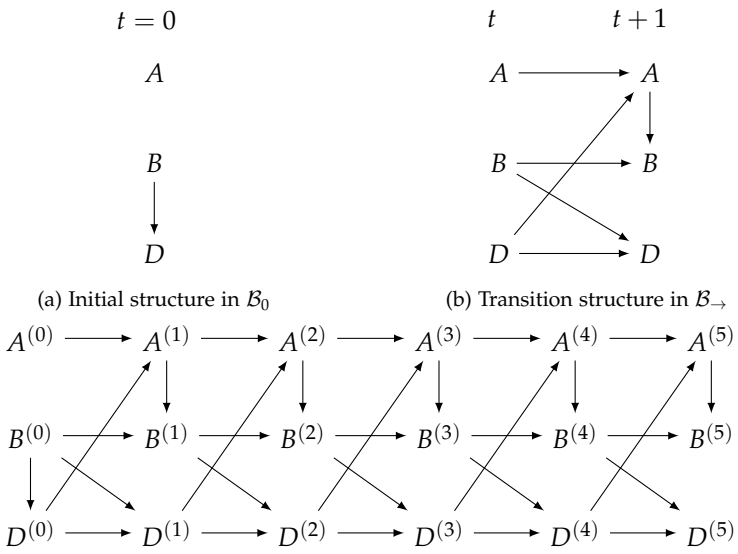
where in Equation 2.18 it is shown that the joint can be written in a modular way based on the factorization provided by the distributions P_0 and $P_{\rightarrow}$. An example of DBN for a medical problem is described in Example 2.2.

Example 2.2. *In a disease process, two symptoms (denoted by A and B) and the administered drug quantity (denoted by D) are observed at regular time intervals for each patient. A DBN is used to model patient evolution, where the structure of the initial model* $\mathcal{B}_0$ *and the transition model* $\mathcal{B}_{\rightarrow}$ *are shown at the top of Figure 2.2. From* $\mathcal{B}_0$ *and* $\mathcal{B}_{\rightarrow}$*, an unrolled DBN over six time points can be obtained, as shown at the bottom of Figure 2.2.*

In the transition model of a DBN the set of arcs from a variable at time t to a variable at time $t + 1$ are often called *intra-temporal* arcs, e.g., the arcs $B^{(0)} \rightarrow D^{(0)}$ and $A^{(1)} \rightarrow B^{(1)}$ in Example 2.2. On the other hand, arcs between variables from the same time point are called *inter-temporal* arcs, such as $A^{(0)} \rightarrow A^{(1)}$ in this example.

2.4.2 *Learning*

Learning DBNs is to a considerable extent similar to learning (static) Bayesian networks. Let us consider a training set D of m i.i.d. sequences, where the jth sequence has observations of the form $\mathbf{x}[j]^{(0)}, \ldots, \mathbf{x}[j]^{(m_j)}$. For convenience, we denote by D_0 the initial slices of D, which amount to m observations, whereas we denote by $D_{\rightarrow}$ the transition instances of D, which amount to m' observations, where $m' = \sum_{i=1}^{m} m_i$. The initial model $\mathcal{B}_0$ and the transition model $\mathcal{B}_{\rightarrow}$ are learned from D_0 and $D_{\rightarrow}$ respectively.



(a) Initial structure in $\mathcal{B}_0$

(b) Transition structure in $\mathcal{B}_{\rightarrow}$

(c) Unrolled DBN for six time points.

Figure 2.2: An example of DBN for a disease process. The CPTs of $\mathcal{B}_0$ and $\mathcal{B}_{\rightarrow}$ are not shown.

In an MLE approach, by a similar reasoning as done with static Bayesian networks (Section 2.3.1) it can be shown [68] that the BIC of a DBN $(\mathcal{B}_0, \mathcal{B}_{\rightarrow})$ with structure $\mathcal{G} = (\mathcal{G}_0, \mathcal{G}_{\rightarrow})$ is given by:

$$\text{BIC}(\mathcal{G} : D) = \text{BIC}_0 + \text{BIC}_{\rightarrow} \tag{2.19}$$

where

$$\text{BIC}_0 = -2 \cdot \log \mathcal{L}(\hat{\theta}_{\mathcal{G}_0} : D_0) + K_0 \cdot \log m \tag{2.20}$$

and

$$\text{BIC}_{\rightarrow} = -2 \cdot \log \mathcal{L}(\hat{\theta}_{\mathcal{G}_{\rightarrow}} : D_{\rightarrow}) + K_{\rightarrow} \cdot \log m' \tag{2.21}$$

such that K_0 and $K_{\rightarrow}$ denote the number of parameters of the initial and transition models respectively.

Equation 2.21 in fact uses the conditional log-likelihood of the transition instances, which is given by $\log \mathcal{L}(\hat{\theta}_{\mathcal{G}_{\rightarrow}} : D_{\rightarrow}) = \sum_{j=1}^{m} \sum_t \log P(\mathbf{x}[j]^{(t+1)} \mid \mathbf{x}[j]^{(t)})$. By maximizing the BIC of $\mathcal{B}_0$ and the BIC of $\mathcal{B}_{\rightarrow}$ independently, the BIC of the complete DBN is maximized as well.

2.5 HIDDEN MARKOV MODELS

In several situations, we might be interested in modeling latent (or hidden) variables, which allow for capturing unmeasured quantities related to the observed

quantities [178]. This might provide improved understanding of the problem at hand, along with other potential advantages such as simplified model structure [67] and better model fit [179].

In this section, we discuss hidden Markov models (HMMs, for short), which can be seen as instances of DBNs from a representation perspective. We focus on several representation aspects of HMMs, while learning is covered in the next section.

2.5.1 *Model architectures*

In a general problem setting, we denote by $\mathbf{X} = \{X_1, \ldots, X_n\}$ the set of observable features, and we assume that there is a set of state variables $\mathbf{S} = \{S_1, \ldots, S_\ell\}$ that we do not observe and are involved in the generation of $\mathbf{X}$ over time. In such problem, we are interested in a temporal model that can be constructed and used feasibly, yet is realistic and insightful. To this end, different sets of assumptions are very often used, taking also into account domain characteristics. As a consequence, the variety of existing HMMs renders different probabilistic interactions between $\mathbf{X}$ and $\mathbf{S}$ (by interaction we refer to unconditional probabilistic dependence).

A general HMM framework is illustrated in Figure 2.3 [25], where the exact form of interactions within states and within observables is abstracted. We start by defining the HMM which captures the interactions denoted by solid lines in Figure 2.3 and can be seen as a basis for several other HMMs.

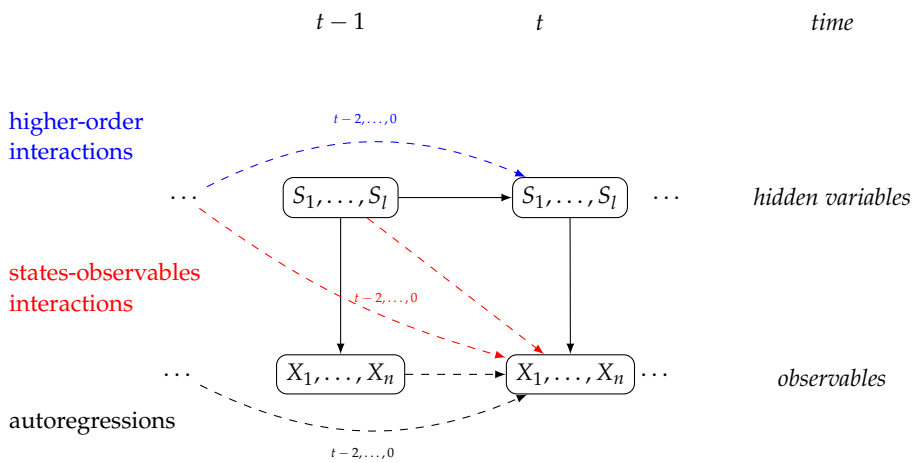


Figure 2.3: An abstracted general HMM with hidden variables $\{S_1, \ldots, S_\ell\}$ and observables $\{X_1, \ldots, X_n\}$. Solid arcs indicate interactions present in the independent HMM and related models.

Definition 2.6 (Hidden Markov model). *A hidden Markov model is a Markovian time-homogeneous system $\lambda = (A, B, v)$ over $\{S, \mathbf{X}\}$, where:*

- $A = P(S^{(t+1)} \mid S^{(t)})$ *is the transition distribution*
- $B = P(\mathbf{X}^{(t)} \mid S^{(t)})$ *is the emission distribution*
- $v = P(S^{(0)})$ *is the initial state distribution*

and $dom(S) = \{s_1, \ldots, s_k\}$ *is called the state space of the model.*

The above definition is based on those of dynamic systems given in Section 2.4, except that in an HMM we repeatedly measure not only observables $\mathbf{X}$, but also a latent variable S. In this case, there is a single latent variable per time point, hence $\mathbf{S} = \{S\}$. It is customary to view A as a matrix $[a_{ij}]$, B as a set $\{b_j(\mathbf{k})\}_{s_j \in \mathrm{dom}(S)}$, and v as a vector $[v(s_i)]$, where:

$$a_{ij} = P(S^{(t+1)} = s_j \mid S^{(t)} = s_i) \tag{2.22}$$

$$b_j(\mathbf{k}) = P(\mathbf{X}^{(t)} = \mathbf{x_k} \mid S^{(t)} = s_j) \tag{2.23}$$

$$v(i) = P(S^{(0)} = s_i) \tag{2.24}$$

The above notation will be useful when describing HMM learning (see Section 2.6.2). By unrolling an HMM over a finite time horizon $\{0, \ldots, T\}$, and from the given assumptions and definitions, the joint distribution of an HMM is:

$$P(\mathbf{X}^{(0:T)}, S^{(0:T)}) = P(S^{(0)}) \prod_{t=0}^{T} P(\mathbf{X}^{(t)} \mid S^{(t)}) \prod_{t=0}^{T-1} P(S^{(t+1)} \mid S^{(t)}) \tag{2.25}$$

A well-known class of HMMs is the *independent* HMM (HMM-I, for short) [102, 138, 141], in which the observables at a given time point are assumed conditionally independent given the state. This additional assumption means that $P(X_i^{(t)} \mid X_j^{(t)}, S^{(t)}) = P(X_i^{(t)} \mid S^{(t)})$ whenever $P(X_j^{(t)}, S^{(t)}) > 0$, for all $t \geq 0$ and $i \neq j$.

Based on the previous assumptions, the joint distribution of an HMM-I is as follows:

$$P(\mathbf{X}^{(0:T)}, S^{(0:T)}) = P(S^{(0)}) \prod_{t=0}^{T} \prod_{i=1}^{n} P(X_i^{(t)} \mid S^{(t)}) \prod_{t=0}^{T-1} P(S^{(t+1)} \mid S^{(t)}) \tag{2.26}$$

2.5.2 *Families of HMMs*

By relaxing the assumptions of the independent HMM based on the general architecture shown in Figure 2.3, different families of HMMs can be derived, as summarized in Figure 2.4. The emissions of a *standard* HMM can be defined as:

$$P(\mathbf{X} \mid S) = \prod_{i=1}^{n} P(X_i \mid S, \pi^{-}(X_i)) \tag{2.27}$$

where $\pi^{-}(X_i)$ denotes the set of parents of X_i excluding the state, and it depends on the Bayesian network associated to the feature space. In the literature, the standard HMM is also known as HMM/BN [119].

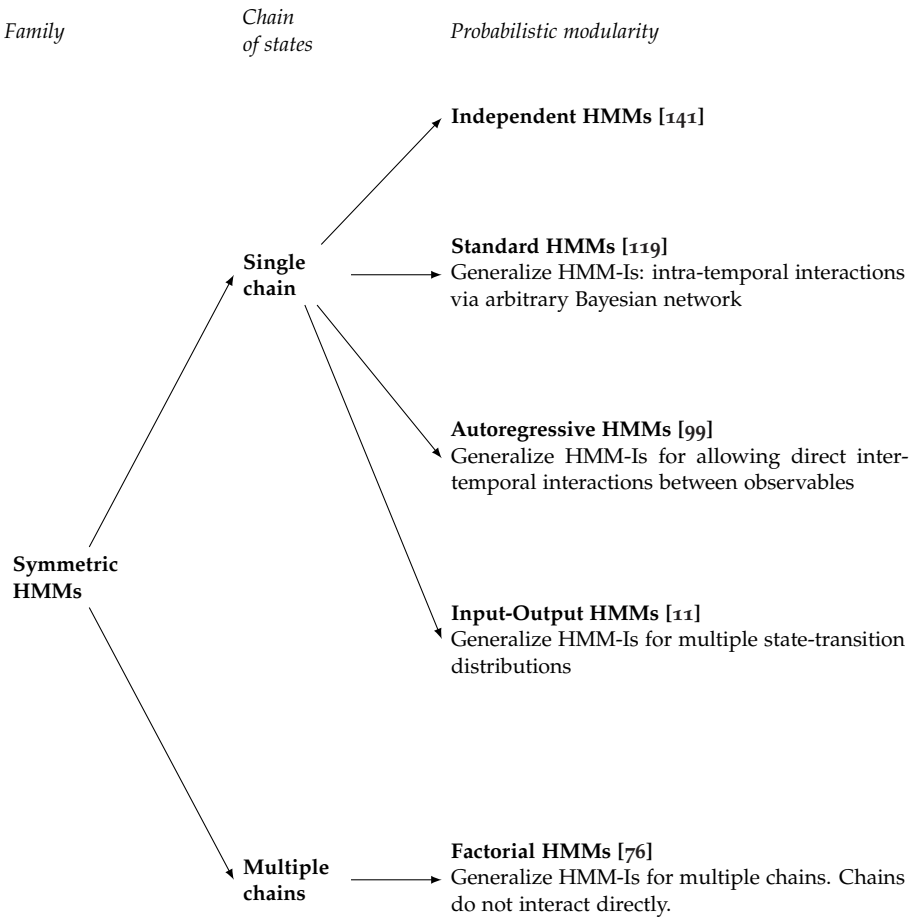


Figure 2.4: HMM families, where *intra-temporal* interactions refer to probabilistic interaction among observables from the same time point, while *inter-temporal* interactions refer to interactions between distinct time points.

The models from Figure 2.4 can be distinguished based on a number of factors, where probabilistic modularity, i.e. the sets of variables that are directly connected in the model, has a major importance. For example, the very high modularity of HMM-Is requires that the state variable summarize more history information, which can imply larger state spaces [76], as opposed to autoregressive HMMs, where the direct interaction between observables is prone to reduce the burden over the hidden states. On the other hand, as autoregressive HMMs might require more parameters, they are prone to be less stable. This illustrates that there can be several trade-offs when a decision must be made about the design of an HMM.

2.5.3 *Learning*

Learning HMMs often reduces to parameter learning, because the structure of models such as the independent HMM and input-output HMM is defined beforehand. However, structure learning might be needed, e.g., for learning standard HMMs and autoregressive HMMs if one wants to decide which autoregressions should be present based on the data. Due to the presence of latent variables, learning HMMs is addressed differently than discussed so far and will be covered in more detail in Section 2.6.

2.6 LEARNING WITH LATENT VARIABLES

In many real-life situations the data might have variables with missing values due to several reasons, e.g., when patients drop out of treatment, do not carry out a measurement or forget to register the result of a measurement, or when a sensor breaks down. At other times, we might be interested in modeling latent (or hidden) variables, i.e., the situation when no values have been observed for a variable, which is the case of probabilistic models such as HMMs. Dealing with missing values or with situations when all values of a variable are missing is done by similar algorithms. In this section we discuss learning of models with latent variables.

Learning Bayesian networks with latent variables is more difficult than the case of complete data, as closed-form solutions are in general not available [15, 21]. Iterative methods are often used, including gradient-based methods [104] and the expectation maximization algorithm (EM, for short) [53, 65, 67]. More recently, research has shown that in some situations parameter learning of Bayesian networks with latent variables can be done in closed form [21], while structure learning has been done by approximating predictive distributions, also in an iteration-free approach [143].

2.6.1 *The expectation-maximization algorithm*

In the situation of latent variables, the space of variables is given by a set of observables $\mathbf{X}$ together with a set of latent variables $\mathbf{S}$. Then, the log-likelihood function of model parameters θ given a probabilistic structure $P(\mathbf{X}\colon \theta)$ and data on $\mathbf{X}$ can be written as:

$$\log \mathcal{L}(\theta\colon \mathbf{x}) = \log P(\mathbf{x}\colon \theta) = \log \sum_{\mathbf{s}} P(\mathbf{x}, \mathbf{s}\colon \theta) \tag{2.28}$$

The presence of the summation inside the logarithm makes it difficult to optimize Equation 2.28 analytically, because the marginal $P(\mathbf{x}\colon \theta)$ might not belong to the exponential family even if $P(\mathbf{x}, \mathbf{s}\colon \theta)$ does [15]. The expectation maximization algorithm is an alternative approach for finding the maximum likelihood in an iterative way. In EM, we refer to $\mathbf{X}$ as the *incomplete data*, and $\mathbf{X} \cup \mathbf{S}$ as the *complete data*. However, as we do not have access to the complete data, EM

resorts to the expected likelihood of the complete data. A general description of the EM procedure is provided in Algorithm 1. In the context of (dynamic) Bayesian networks, the described EM algorithm assumes the network structure is known and the goal is to perform parameter estimation, thus θ refers to model parameters.

Algorithm 1 Expectation-maximization algorithm.

Input: D: a set of data points of the form $\mathbf{X} = \{X_1, \ldots, X_n\}$; $\mathbf{S} = \{S_1, \ldots, S_\ell\}$: a set of latent variables.
Output: θ: model parameters for the distribution $P(\mathbf{x}, \mathbf{s})$.

1: Choose initial model parameters θ^{old}.
2: **repeat**
3: **E step**: Compute $P(\mathbf{s} \mid \mathbf{x}\colon \theta^{\text{old}})$
4: **M step**: Compute θ^{new} as follows:

$$\theta^{\text{new}} \leftarrow \arg\max_{\theta} Q(\theta, \theta^{\text{old}}) \tag{2.29}$$

$$\text{where} \quad Q(\theta, \theta^{\text{old}}) \stackrel{\text{def}}{=} \mathbb{E}_{\mathbf{s}}\left[\log P(\mathbf{x}, \mathbf{s}\colon \theta) \mid \mathbf{x}\colon \theta^{\text{old}}\right] \tag{2.30}$$

$$= \sum_{\mathbf{s}} P(\mathbf{s} \mid \mathbf{x}\colon \theta^{\text{old}}) \log P(\mathbf{x}, \mathbf{s}\colon \theta) \tag{2.31}$$

$$\theta^{\text{old}} \leftarrow \theta^{\text{new}} \tag{2.32}$$

5: **until** convergence of the log-likelihood or the model parameters is achieved
6: **return** θ^{new}

The goal of Algorithm 1 is to maximize the expected log-likelihood with regard to the latent states (Equation 2.30) in an iterative way. Algorithm 1 starts with initial model parameters θ^0, which is often randomly generated. Then, EM calculates the term of Equation 2.30 regarding the current parameters, i.e. $P(\mathbf{s} \mid \mathbf{x}\colon \theta^{\text{old}})$, which is known as the *E step*. Once this is done, it is possible to evaluate candidate model parameters for the expected log-likelihood. The goal now is to find new model parameters that optimizes this expectation, which is known as the *M step*. The process of generating new model parameter estimates from the current one is repeated until no improvement is possible, which means that a local maxima or a saddle point of the likelihood function has been achieved [65]. It is possible to show [15] that each cycle of E and M steps generates a new model that is at least as good as the previous one.

2.6.2 *The Baum-Welch algorithm*

Models with latent variables such as HMMs are often learned by means of the EM algorithm, which is also known as the Baum-Welch algorithm in this context [141]. In this section, we consider the learning of HMMs with fixed structure, for which learning reduces to parameter learning. We use independent HMMs to illustrate the necessary calculations. These involve estimating the parameters θ, which include the emission distributions $P(X_i^{(t)} \mid S^{(t)})$, with $X_i \in \mathbf{X}$ and $\mathrm{dom}(S) = \{s_1, \ldots, s_k\}$, the transitions $P(S^{(t+1)} \mid S^{(t)})$ and the initial probabilities $P(S^{(0)})$.

We consider the same learning setting as that of learning DBNs (Section 2.4.2), but in order to simplify the notation, we consider that the data D consists of a single sequence over $\{0, \ldots, T\}$. This can be easily extended to the case of multiple i.i.d. sequences. Let us denote by θ^{old} the parameters of current HMM-I and by θ the parameters of new HMM-I to be obtained. The Q function of Equation 2.30 for HMMs becomes:

$$Q(\theta, \theta^{\text{old}}) = \sum_{s^{(0:T)}} P(s^{(0:T)}, \mathbf{x}^{(0:T)} \colon \theta^{\text{old}}) \cdot \log P(\mathbf{x}^{(0:T)}, s^{(0:T)} \colon \theta) \tag{2.33}$$

For convenience, in Equation 2.33 the joint $P(s, \mathbf{x} \colon \theta^{\text{old}})$ was used. It holds that $P(s, \mathbf{x} \colon \theta^{\text{old}}) = P(\mathbf{x} \colon \theta^{\text{old}}) P(s \mid \mathbf{x} \colon \theta^{\text{old}})$, thus one can use either the joint or the conditional probability for this development because they are independent of θ. By substituting the joint of HMMs (Equation 2.26) into Equation 2.33, we obtain:

$$\begin{aligned} Q(\theta, \theta^{\text{old}}) = & \sum_{s^{(0:T)}} P(s^{(0:T)}, \mathbf{x}^{(0:T)} \colon \theta^{\text{old}}) \cdot \log P(s^{(0)}) \\ & + \sum_{s^{(0:T)}} P(s^{(0:T)}, \mathbf{x}^{(0:T)} \colon \theta^{\text{old}}) \left(\sum_{t=0}^{T} \sum_{i=1}^{n} \log P(X_i^{(t)} \mid s^{(t)}) \right) \\ & + \sum_{s^{(0:T)}} P(s^{(0:T)}, \mathbf{x}^{(0:T)} \colon \theta^{\text{old}}) \left(\sum_{t=0}^{T-1} \log P(s^{(t+1)} \mid s^{(t)}) \right) \end{aligned} \tag{2.34}$$

In order to maximize such Q function, one can maximize each term independently. The term referring to the initial distribution can be written as:

$$\sum_{s^{(0:T)}} P(s^{(0:T)}, \mathbf{x}^{(0:T)} \colon \theta^{\text{old}}) \cdot \log P(s^{(0)}) \tag{2.35}$$

$$= \sum_{s^{(0:T)}} P(s^{(0:T)}, \mathbf{x}^{(0:T)} \colon \theta^{\text{old}}) \cdot \log v(i) \tag{2.36}$$

By setting the derivative with respect to $v(i)$ to zero, and introducing the Lagrange α multiplier to ensure the constraint $\sum_{i=1}^{k} v(i) = 1$ we obtain:

$$\frac{\partial}{\partial v} \left(\sum_{s^{(0:T)}} P(s^{(0:T)}, \mathbf{x}^{(0:T)} \colon \theta^{\text{old}}) \cdot \log v(i) - \alpha \left(\sum_{i=1}^{k} v(i) - 1 \right) \right) = 0 \tag{2.37}$$

It is possible to show [14] that this results in:

$$\overline{v}(i) = \frac{P(\mathbf{x}^{(0:T)}, s_i^{(0)} : \theta^{\text{old}})}{P(\mathbf{x}^{(0:T)} : \theta^{\text{old}})} \tag{2.38}$$

By similar reasoning we obtain the transitions and emissions:

$$\overline{a}_{ij} = \frac{\sum_{i=0}^{T-1} P(\mathbf{x}^{(0:T)}, s_i^{(t)}, s_j^{(t+1)} : \theta^{\text{old}})}{\sum_{i=0}^{T-1} P(\mathbf{x}^{(0:T)}, s_i^{(t)} : \theta^{\text{old}})} \tag{2.39}$$

$$\overline{b}_j(\mathbf{k}) = \frac{\sum_{i=0}^{T} P(\mathbf{x}^{(0:T)}, s_j^{(t)} : \theta^{\text{old}}) \cdot \mathbb{1}(\mathbf{k})}{\sum_{i=0}^{T} P(\mathbf{x}^{(0:T)}, s_j^{(t)} : \theta^{\text{old}})} \tag{2.40}$$

where $\mathbb{1}(\mathbf{k}) = 1$ if $\mathbf{X}^{(t)} = \mathbf{x}_\mathbf{k}$, and 0 otherwise.

In order to actually compute the above probabilities, some useful quantities are defined:

$$\gamma_t(i) \stackrel{\text{def}}{=} P(s_i^{(t)} \mid \mathbf{x}^{(0:T)} : \theta^{\text{old}}) \tag{2.41}$$

$$\xi_t(i,j) \stackrel{\text{def}}{=} P(s_i^{(t)}, s_j^{(t+1)} \mid \mathbf{x}^{(0:T)} : \theta^{\text{old}}) \tag{2.42}$$

In many families of HMMs, the γ and ξ values can be efficiently computed by the forward-backward procedure based on dynamic programming [14, 15]. This fact is important for keeping the EM iterations efficient. Then, it is possible to show [141] that parameter update leads to $\theta^{\text{new}} = (\overline{A}, \overline{B}, \overline{v})$ given by:

$$\overline{v}(i) = \gamma_0(i) \tag{2.43}$$

$$\overline{a}_{ij} = \frac{\sum_{t=0}^{T-1} \xi_t(i,j)}{\sum_{t=0}^{T-1} \gamma_t(i)} \tag{2.44}$$

$$\overline{b}_j(\mathbf{k}) = \frac{\sum_{t=0}^{T} \gamma_t(j) \cdot \mathbb{1}(\mathbf{k})}{\sum_{t=0}^{T} \gamma_t(j)} \tag{2.45}$$

2.6.3 *Number of latent states*

One important hyperparameter of HMMs is the number of latent states. It is often the case that this hyperparameter is not known in advance, hence one can resort to estimating it using data. Information-theoretic criteria, such as AIC and BIC, can be used for this task [35, 138], with the advantage of low computational cost, something that is important for tasks as Bayesian-network structure learning.

Selecting the number of states can also be done by means of cross validation. While cross validation is typically more costly than penalized scoring functions, with cross validation one is able to directly estimate the performance of the model on new data, which allows for dealing with overfitting. A particular advantage of cross validation in HMMs is that it avoids assumptions made by penalized scoring functions, which require that the actual distribution of the data belongs to one of the models being compared [35].

One approach for selecting the number of states by means of cross validation is by incrementally increasing the number of states until model generalization stabilizes or deteriorates [102, 144]. To evaluate a model with q states by means of k-fold cross-validation, the training data is obtained from $(k-1)/k$ percent of sequences, while the validation data is obtained from the remaining sequences. The average log-likelihood of validation data is reported as the performance of the model, which can then be compared against, e.g., a model with $q+1$ states.

2.6.4 *Structure learning with missing data*

It is often the case that models with latent variables also need to have some of their structure estimated from data. For example, more general HMMs which adopt less restrictive assumptions on the probabilistic interaction among variables (see Figure 2.4) might require structure learning on the emission space, if there is no *a priori* domain knowledge about typical interactions. Using an inadequate or too restrictive model structure can considerably limit model emissions, which has been considered important for allowing HMMs to properly capture complex distributions [12, 102, 123] since it governs how observations are emitted to the external world.

In the situation of learning with latent variables and unknown structure, one can resort to an extension of the EM algorithm called *structural EM* (SEM, for short) [65, 67]. This is the case of Bayesian networks with latent variables and unknown structure, as well as standard HMMs (Equation 2.27), where the emission distribution follows an arbitrary Bayesian network, thus the observable variables might not be independent given the state.

In the following development, we consider that the model to be learned is a Bayesian network with structure $\mathcal{G}$ and parameters θ (for convenience, we use θ instead of P in this algorithm). The SEM procedure is described as follows:

1. Choose $\mathcal{B}^0 = (\mathcal{G}^{(0)}, \theta^{(0)})$ randomly
2. Loop for $n = 0, 1, \ldots$ until convergence:
   a) Find $\mathcal{G}^{\text{new}}$, as follows:

$$\mathcal{G}^{\text{new}} \leftarrow \arg\max_{\mathcal{G}} Q((\mathcal{G}, \theta_{\mathcal{G}}), (\mathcal{G}^{\text{old}}, \theta^{\text{old}})) \tag{2.46}$$

   where

$$Q(\mathcal{B}, \mathcal{B}^{\text{old}}) \stackrel{\text{def}}{=} \mathbb{E}_s \left[\log P(\mathbf{x}, s \colon \mathcal{B}) - \text{Pen}(\mathcal{B}, \mathbf{x}) \mid \mathbf{x}, \mathcal{B}^{\text{old}} \right] \tag{2.47}$$

b) Let $\theta^{\text{new}} \leftarrow \arg\max_\theta Q((\mathcal{G}^{\text{new}}, \theta), (\mathcal{G}^{\text{old}}, \theta^{\text{old}}))$

One important insight of SEM is placing structure learning inside EM calculations. In the SEM procedure, each iteration involves structure learning and parameter learning, both based on expected counts. The expected counts are based on probabilities computed over the current model $\mathcal{B}^{\text{old}}$. In step(a), structure learning is executed based on the expected counts of the current parameters θ^{old}. As SEM involves selecting model structure, an additional factor is included in the expected score to penalize for model complexity. Once this is done, in step(b) new parameters θ^{new} are computed for this structure $\mathcal{G}^{\text{new}}$. Note that step(b) can be seen as the parametric part of SEM.

It is possible to show [65] that in order to guarantee convergence it is sufficient to find a BN $\mathcal{B}^{\text{new}}$ with improved score compared to the BN found on the previous SEM iteration, instead of maximizing the expected score shown above. This is useful because it is often the case that heuristic search is used for obtaining $\mathcal{B}^{\text{new}}$, which does not guarantee that in each iteration the expected score is maximized.

3

ASYMMETRIC HIDDEN MARKOV MODELS

In this chapter, we introduce asymmetric hidden Markov models, which generalize the emission distributions of HMMs to arbitrary Bayesian-network distributions, allowing for state-specific graphical structures defined over the feature space. As a consequence, HMM-As are able to render more compact state spaces, thus from a learning perspective HMM-As can better cope with model overfitting compared to other HMM architectures. We study representation properties of asymmetric and symmetric HMMs, as well as provide a learning algorithm for HMM-As. Empirical results based on simulated and real-world data from several domains show the effectiveness of modeling more general asymmetries as done by HMM-As and the insight that such models can yield.

3.1 INTRODUCTION

In many dynamic systems, complex patterns of observations are emitted over time. It is often the case that parts of the underlying process are not observed, e.g. because it is too difficult or impossible. This situation imposes challenges for capturing the interactions between observable features. Hidden Markov models are often employed as models for dynamic systems, having been successful in speech recognition and synthesis domain [119, 141, 168]. HMMs have also been applied to problems such as gene prediction and biological sequences [60, 165], information retrieval [64, 155], and business processes [148]. However, it has been also recognized that HMMs might face limitations to properly capture distributions when limited data is available [13, 75, 119]. Furthermore, HMMs in practice often have a single chain of states and impose a naive structure over the feature space, which on the one hand alleviates learning and inference costs, but on the other hand gives rise to larger state spaces that can lead to learning issues (e.g. model overfitting) and unsatisfactory problem insight.

Research has been dedicated to extending HMMs for representing more structural information, aiming to render more useful and accurate models, e.g. factorial HMMs [76], hierarchical HMMs [62], HMM/BN [119], and autoregressive HMMs [139]. Nevertheless, these extensions do not capture more specialized independences, often referred to as asymmetric independences or local structure [73], i.e. independences that hold for subsets of values of the involved variables. In the context of graphical models, the representation of asymmetries dates back

to Bayesian multinets [73] and similarity networks [84], and had its importance recognized for allowing better probabilistic inference [19, 32, 173], learning [66, 137], and for improving problem insight [102] as well.

In the context of HMMs, however, research on capturing asymmetries has been much more limited, with a focus mostly on autoregressive models. Such models include the representation of higher-order autoregressive interactions by means of dynamic multinets [12], tree-based interactions on the observables space by means of Chow-Liu HMMs [102], and a combination of first-order autoregressions and tree-based interactions as implemented by conditional Chow-Liu HMMs [12]. Therefore, a model able to capture more general asymmetries on the observables space is needed. The literature also lacks a better understanding of the implications of employing asymmetry models in time series settings. To address these research aspects, we propose *asymmetric hidden Markov models* (HMM-As). In HMM-As observations are emitted according to state-specific Bayesian-network distributions, thus these models are able to represent independences that are not represented in symmetric HMMs.

The contributions of this chapter are as follows. We first define HMM-As, and compare its representation aspects with families of symmetric HMMs with respect to their state space dimensions. Then, we discuss a learning algorithm for HMM-As, which is based on the structural expectation-maximization framework [53, 65], and additionally analyze computational costs associated to symmetric and asymmetric HMMs. A set of varied simulations is then presented, with special attention to the effect of different dataset sizes and number of underlying structured states when learning symmetric and asymmetric models. Finally, we discuss experiments based on real-world datasets, where we take a close look at the obtained models in order to gain additional insight supported by HMM-As. Such empirical results indicate that HMM-As can be successfully used to obtaining new insight from real-life problems from several domains, including business processes, monitoring of urban pollution, and industrial processes.

The remainder of this chapter is organized as follows. In Section 3.2 we provide basic notions on distribution asymmetries and HMMs that represent asymmetries. In Section 3.3 we define HMM-As and relate them to other HMMs. In Section 3.4 a learning procedure for HMM-As is introduced. Section 3.5 reports results based on simulated data, while Section 3.6 reports results based on real-world data and discusses problem insight. In Section 3.7 the related work is discussed. The summary and future work are discussed in Section 3.8.

3.2 BASIC NOTIONS

In Chapter 2, we discussed different classes of HMMs, which included the most common one, i.e., the independent HMM, as well as other HMMs. It is worth noting that the HMMs shown in Figure 2.4 do not capture *asymmetries* in the distribution, i.e. independences that are valid for some values within the domains of the variables. Such independences can be formally defined by the notion of *context-specific independences*, which we define next, based on [19].

Definition 3.1 (Contextual independence)**.** *Let P be a probability distribution over the sets of random variables $\mathbf{V}$, $\mathbf{W}$, $\mathbf{U}$, and $\mathbf{C}$, which are pairwise disjoint. We say that $\mathbf{V}$ is contextually independent of $\mathbf{W}$ given $\mathbf{U}$ and the context $\mathbf{c} \in dom(\mathbf{C})$ if $(\mathbf{V} \perp\!\!\!\perp_P \mathbf{W} \mid \mathbf{c}, \mathbf{U})$ for all values of $\mathbf{V}$, $\mathbf{W}$, and $\mathbf{U}$.*

Context-specific independences are able to capture independence statements that are not captured with conditional independence statements. In the context of HMMs, the context is typically given by values of the state variables(s), and we shall refer to such statements as *asymmetric independences* in this chapter.

A summary of HMMs that represent distribution asymmetries is given in Table 3.1. For such HMMs, each state $s \in \mathrm{dom}(S)$ determines the parent set of each observable, thus leading to asymmetric independences of the form:

$$(\mathbf{V}^{(t)} \perp\!\!\!\perp_P \mathbf{W}^{(t)} \mid s^{(t)}, \mathbf{U}) \tag{3.1}$$

where $\mathbf{V}, \mathbf{W} \subseteq \mathbf{X}$. The set $\mathbf{U}$ depends on the state s at t, and it determines the model architecture. For example, in Chow-Liu HMMs $\mathbf{U} \subseteq \mathbf{X}^{(t)}$, whereas in dynamic multinets $\mathbf{U} \subseteq \mathbf{X}^{(0:t-1)}$.

Model	Chain of states	Distribution asymmetries	Learning
Dynamic Multinet [12]	Single	Higher-order autoregressions between observables. No intra-temporal correlations.	Discriminative (classification)
Chow-Liu HMM [102]	Single	Intra-temporal interactions modeled by tree distributions.	Generative
Conditional Chow-Liu HMM [102]	Single	First-order autoregressions and tree-based intra-temporal interactions.	Generative
Activator DBN [123]	No chain	Autoregressions and intra-temporal interactions between observables.	Not available
Asymmetric HMM (this chapter)	Single	Intra-temporal interactions modeled by arbitrary Bayesian network distributions.	Generative

Table 3.1: HMM families which represent independence asymmetries.

Representing distribution asymmetries is important for inference and learning, however, as Figure 2.4 and Table 3.1 show, research to represent asymmetries has been much narrower in the context of HMMs. This is justified by the sequential nature and the role played by hidden states in HMMs, which imposes other challenges when compared to the static case. We further discuss work on representing distribution asymmetries in HMMs as follows.

As Table 3.1 indicates, systematic approaches for learning Chow-Liu HMMs are available by means of a generative-based learning. However, the representation of state-specific asymmetries in such HMMs is limited to trees, which can be

harmful especially when the feature space has more features, and thus many more structures become available. On the other hand, dynamic multinets directly model potentially longer-history correlations by means of autoregressions. Yet, no instantaneous (i.e. intra-temporal) interactions are captured, which makes them closer to the original ideas of autoregressive HMMs [99, 139] by not fully exploring the graphical structure.

The learning approach of the previous asymmetry-aware HMMs is targeted at specific tasks, namely, classification. Thus, there is a need for models that can represent more general asymmetries within the feature space, yet in a compact manner to avoid the need for large amounts of data. Furthermore, the literature lacks a better understanding of the representation capacities of the independent HMM and other, structured HMMs with respect to state space dimensions and model fit when the data generation process has varying amount of structure.

3.3 ASYMMETRIC HIDDEN MARKOV MODELS

Asymmetric hidden Markov models generalize HMMs by allowing the emission distributions to represent additional qualitative independence per state. In the following we define HMM-As by first defining the association between states and Bayesian-network distributions, followed by a discussion on model parameterization.

3.3.1 *Model specification*

In order to define asymmetries in HMMs, we consider that hidden states induce local models over the observables. This notion can be conveniently represented by conditional Bayesian networks [104], in which a distribution $P(\mathbf{X} \mid S)$ is defined for the observables $\mathbf{X}$ and the state S. As standard conditional BNs provide a single factorization of $\mathbf{X}$ for all $s \in \mathrm{dom}(S)$, we extend this notion for accommodating more general state-specific models as follows.

Definition 3.2 (State-specific Bayesian network)**.** *Let $\mathbf{X}$ and S be random variables. For each $s \in dom(S)$, we associate a Bayesian network over $\mathbf{X}$ called state-specific Bayesian network for s. If $B_s = (P_s, G_s)$ is the state-specific BN associated to s, we define the following conditional distribution:*

$$P(\mathbf{X} \mid s) = P_s(\mathbf{X}) \tag{3.2}$$

$$= \prod_{i=1}^{n} P_s(X_i \mid \pi_s(X_i)) \tag{3.3}$$

where $\pi_s(X_i)$ denotes the parent set of X_i as dictated by the state-specific BN $B_s = (G_s, P_s)$, in which G_s denotes its graphical structure and P_s its conditional probability tables.

The previous definitions map hidden states to BNs, thus conveniently allowing multiple sets of parents for the features in $\mathbf{X}$, one for each state-specific BN.

Definition 3.3 (Asymmetric hidden Markov model)**.** *An asymmetric hidden Markov model over the random variables* $(\mathbf{X}, S)$ *is a dynamic system* $\lambda = (M_{\rightarrow}, M_{\downarrow}, M_0)$*, where* M_0 *is an initial distribution* $P(S^{(0)})$*,* $M_{\rightarrow}$ *is a transition distribution* $P(S^{(t+1)} \mid S^{(t)})$*, and* $M_{\downarrow}$ *is an emission distribution given by*

$$P(\mathbf{X}^{(t)} \mid S^{(t)}) = P_{S^{(t)}}(\mathbf{X}^{(t)}) \tag{3.4}$$

From the definitions shown above, HMM-As are able to capture more qualitative independences in their topology than HMMs. Yet, HMM-As will share a few assumptions with HMMs, namely: the Markovian property and time-invariance. A third assumption that will also hold in HMM-As establishes that the intertemporal interaction between features must occur via state variables. Hence, given these assumptions, an unrolled HMM-A over the time horizon $\{0, \ldots, T\}$ has the following joint distribution:

$$\begin{aligned} P(S^{(0:T)}, \mathbf{X}^{(0:T)}) = P(S^{(0)}) \prod_{t=0}^{T-1} P(S^{(t+1)} \mid S^{(t)}) \\ \cdot \prod_{t=0}^{T} \prod_{i=1}^{n} P_{S^{(t)}}(X_i^{(t)} \mid \pi_{S^{(t)}}(X_i)) \end{aligned} \tag{3.5}$$

We note that standard HMMs (see Section 2.5.2) are therefore special cases of HMM-As, since in the standard HMMs every state is associated to the same Bayesian-network structure, i.e. $G_{s_i} = G_{s_j}$ for every $s_i, s_j \in \mathrm{dom}(S)$. An HMM-A can be also visualized as a probabilistic automaton, providing an intuitive representation for states and transitions, as Example 3.1 shows.

Example 3.1. *On a regular basis, measurements of print quality (PQ), room temperature (RT), ink type (IT), and media type (MT) are taken for an industrial printer. An HMM-A* $\mathcal{M}_1$ *for this problem has hidden states that dictate the underlying dynamics, named 'normal', 'failing mode one', and 'failing mode two', denoted by* s_1*,* s_2*, and* s_3 *respectively.* $\mathcal{M}_1$ *is shown in Figure 3.1 as a probabilistic automaton, which runs by alternating taking probabilistic transitions and emitting multivariate observations* $(PQ^{(t)}, RT^{(t)}, IT^{(t)}, MT^{(t)})$ *according to the states which it traverses.*

3.3.2 *Parameterization*

The conditional probability table of each observable X_i in HMMs has the form $P(X_i \mid S, \pi^{-}(X_i))$, where π^{-} refers to the other parents excluding the state. On the other hand, in HMM-As observables have their parameters associated to state-specific BNs, whose CPTs do not explicitly show the states. Nevertheless, CPTs in the standard sense can easily be obtained from HMM-As, as illustrated next.

Example 3.2. *In the HMM-A* $\mathcal{M}_1$ *(see Example 3.1), the conditional probability tables that are rendered for its feature set are shown in Figure 3.2.*

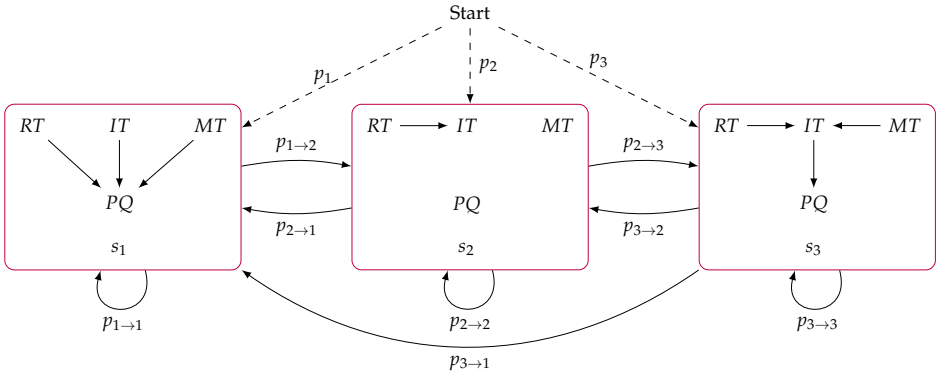


Figure 3.1: Probabilistic automaton representation for HMM-A $\mathcal{M}_1$ (dashed arcs indicate initial transitions; zero probabilities are not shown). State-specific BNs are shown in rounded rectangles.

π	**Parameters**
s_1, RT, IT, MT	$\theta_{PQ\|s_1,RT,IT,MT}$
s_2	$\theta_{PQ\|s_2}$
s_3, IT	$\theta_{PQ\|s_3,IT}$

(a) Node PQ

π	**Parameters**
s_1	$\theta_{RT\|s_1}$
s_2	$\theta_{RT\|s_2}$
s_3	$\theta_{RT\|s_3}$

(b) Node RT

π	**Parameters**
s_1	$\theta_{IT\|s_1}$
s_2, RT	$\theta_{IT\|s_2,RT}$
s_3, RT, MT	$\theta_{IT\|s_3,RT,MT}$

(c) Node IT

π	**Parameters**
s_1	$\theta_{MT\|s_1}$
s_2	$\theta_{MT\|s_2}$
s_3	$\theta_{MT\|s_3}$

(d) Node MT

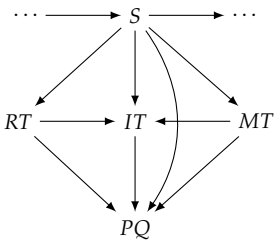
Figure 3.2: Parameterization of probability tables for HMM-A $\mathcal{M}_1$.

Given the HMM-A $\mathcal{M}_1$, it is possible to obtain a standard HMM that represents its distribution over the feature space by turning the asymmetric independences of $\mathcal{M}_1$ into non-asymmetric independences, by taking the minimal directed acyclic graph (DAG) that includes all the dependences of the states in $\mathcal{M}_1$. Let us refer to such a model as *simulating* HMM, which is illustrated next.

Example 3.3. *Let $\mathcal{M}'_1$ be a standard HMM for simulating the HMM-A $\mathcal{M}_1$, such that both models have the same number of states. $\mathcal{M}_1$ includes asymmetric independences such as $(PQ \perp\!\!\!\perp RT \mid s_2)$, which does not hold neither in s_1 nor in s_3. This leads to the conditional dependence $(PQ \not\perp\!\!\!\perp RT \mid S)$, which therefore holds in the simulating HMM $\mathcal{M}'_1$. Similarly, in $\mathcal{M}_1$ it holds that IT and MT are independent in s_1 and s_2 only, hence, it must hold $(IT \not\perp\!\!\!\perp MT \mid S)$ in the simulating HMM. As a consequence, the structure of emissions in $\mathcal{M}'_1$ is denser than the state-specific ones from $\mathcal{M}_1$, as it can be noted from Figure 3.3a showing the emission structure of $\mathcal{M}'_1$.*

It is worth noting that, e.g., the CPT for IT is $P(IT \mid S, RT, MT)$, although direct dependence between IT and $\{RT, MT\}$ exists only when S is s_3 in $\mathcal{M}_1$, which means

that redundancies will exist in this CPT, as shown in Figure 3.3b. Finally, the total number of independent emission parameters in $\mathcal{M}_1$ is 24: 11 from s_1, 5 from s_2, and 8 from s_3. On the other hand, in $\mathcal{M}_1'$ there are 42 independent parameters, obtained by computing the size of the CPT for each variable.



(a) Graphical structure (emissions only).

CPT in $\mathcal{M}_1'$	Parameters in $\mathcal{M}_1$
$P(MT \mid S)$	$\theta_{MT\|S}$
$P(RT \mid S)$	$\theta_{RT\|S}$
$P(IT \mid MT, RT, s_1)$	$\theta_{IT\|s_1}$
$P(IT \mid MT, RT, s_2)$	$\theta_{IT\|s_2,RT}$
$P(IT \mid MT, RT, s_3)$	$\theta_{IT\|s_3,MT,RT}$
$P(PQ \mid RT, IT, MT, s_1)$	$\theta_{PQ\|s_1,RT,IT,MT}$
$P(PQ \mid RT, IT, MT, s_2)$	$\theta_{PQ\|s_2}$
$P(PQ \mid RT, IT, MT, s_3)$	$\theta_{PQ\|s_3,IT}$

(b) Conditional probability tables.

Figure 3.3: Standard HMM $\mathcal{M}_1'$ that simulates the HMM-A $\mathcal{M}_1$.

Two points with further implications follow from Example 3.3. As HMM-As allow for savings in the representation size due to the direct representation of asymmetries in the distribution, one can readily take advantage of these for speeding up probabilistic inference. Secondly, in HMM-As where a few states induce small amounts of dependences (e.g. state s_2 in $\mathcal{M}_1$), the CPTs of the corresponding standard HMM will be large enough to cover the amount of dependences resulting from the union of all state-specific dependences of the original HMM-A. If there is a great disparity in the amount of asymmetries among the states of the HMM-A, the standard HMM will likely require more probabilistic parameters as well. As a consequence, standard HMMs are prone to reveal less insight into practical problems.

3.3.3 *Representation aspects*

In the following, we discuss how standard and independent HMMs can represent HMM-A distributions, and the effects of such procedure on the state space of the former model families.

3.3.3.1 *Relationship with standard HMMs*

We provided in Example 3.3 an intuition on how to obtain a standard HMM able to simulate a particular HMM-A, i.e. an HMM that represents the same distribution over the space of observables. Intuitively, the simulating HMM is prone to have a denser structure compared to the individual states of the original HMM-A, which in the limit reaches a fully connected structure. This is the main idea used in Proposition 3.1 and its proof (see the Appendix) to show that a standard HMM can be obtained from a given HMM-A in the general case. This result also indicates that the simulating HMM does not need additional states for the simulation.

Proposition 3.1. *Let $\mathcal{M}$ be an asymmetric HMM with k states over the observables $\mathbf{X}$, where each $X_i \sim \underline{\text{Multinomial}}$, $i = 1, \ldots, n$. Then, there exists a standard HMM $\mathcal{M}'$ with k states over the same observables which simulates $\mathcal{M}$, i.e. $P'(\mathbf{X}^{(0:T)}) = P(\mathbf{X}^{(0:T)})$, where P and P′ denote the joint distributions of $\mathcal{M}$ and $\mathcal{M}'$ over $\mathbf{X}$ respectively.*

Although the proof of Proposition 3.1 uses an argument based on full connectivity, this is not strictly necessary as the structure on the simulating HMM depends on the amount and form of asymmetries in the original HMM-A. Nevertheless, as Figure 3.3b shows, parameter redundancy at the level of states is likely to occur in the standard HMM, preventing inference from readily benefiting from distribution asymmetries, as such redundancies are encoded in the CPTs, which is not the case in HMM-As.

3.3.3.2 *Relationship with independent HMMs*

While standard HMMs can simulate HMM-As using the same number of states, it is straightforward to see that independent HMMs are not able to do so in the general case. It turns out, however, that the simulation process becomes possible at the cost of expanding the state space of HMM-Is. Intuitively, the more general independence assertions in each state of a given HMM-A must be *broken* into multiple and naively-structured states. We show this result by means of Proposition 3.2.

Proposition 3.2. *Let $\mathcal{M}$ be an asymmetric HMM with k states over the observables $\mathbf{X}$, where each $X_i \sim \underline{\text{Bernoulli}}$, $i = 1, \ldots, n$. Then, there exists an independent HMM $\mathcal{M}'$ with k' states over the same observables, such that $\mathcal{M}'$ simulates $\mathcal{M}$ and $k' \leq k2^n$.*

It is straightforward to extend Proposition 3.2 for the more general case of multinomial observables. The proof of Proposition 3.2 (see Appendix 3.A) provides a method for simulating HMM-A distributions by means of HMM-Is, and it also shows an upper bound on the number of states required by the HMM-I. In practice, the amount of dependences per state and the numerical parameterization of the structured model can greatly vary, hence the number of states that a simulating HMM-I requires tends to be lower than the bound, although it can still be much higher than the original number of states of the original HMM-A. Nevertheless, as we further show in this work, a substantial increase in the

state space can be expected when simulating lowly- and moderately-structured distributions.

3.4 LEARNING

In this section, we present a learning algorithm for HMM-As. We discuss computational costs for this and other families of HMMs as well.

3.4.1 *Learning setting*

In order to learn HMM-As, we assume that state variables are not observed and the graphical structure for emission distributions is unknown. In this case, i.e. learning under missing data and unknown structure, the likelihood function of the observed data is non-decomposable by the graphical structure [15], which makes analytical methods impossible. The structural expectation-maximization (see Section 2.6.4) is often employed in these settings, which serves as a basis for the learning procedure we develop for HMM-As.

The learning setting is score-based and is as follows. Consider a dataset D of m i.i.d. complete sequences, where the ith sequence has the form $\mathbf{x}[i]^{(0)}, \ldots, \mathbf{x}[i]^{(m_i)}$. Given an integer k, we aim to learn an HMM-A with k states that best fits D. As in the structural EM, HMM-A learning is based on the idea of placing structure learning in each cycle of E and M steps. The learning procedure for learning HMM-As is described next, together with a discussion on its cost.

3.4.2 *Expectation step*

In the E step the current model λ^{old} is used for computing two expected statistics: the expected occupancy of each state (denoted by γ), and the expected transitions between any two states (denoted by ξ). For the sake of exposition, we show derivations for the expected statistics considering a single sequence with length $\{0, \ldots, T\}$, which is straightforward to extend for multiple sequences as the sequences are assumed i.i.d. We repeat below the notation of expected statistics given in Section 2.6.2 for convenience:

$$\gamma_t(j) \stackrel{\text{def}}{=} P(S^{(t)} = s_j \mid D\colon \lambda^{\text{old}}) \tag{3.6}$$

$$\xi_t(i,j) \stackrel{\text{def}}{=} P(S^{(t)} = s_i, S^{(t+1)} = s_j \mid D\colon \lambda^{\text{old}}) \tag{3.7}$$

Based on the assumptions regarding the HMM-A topology, it is possible to show that the expected statistics can be given by means of the so-called *forward*

and *backward* variables [141], denoted by α and β respectively, similarly to regular HMMs:

$$\gamma_t(j) = \frac{\alpha_t(j) \cdot \beta_t(j)}{\sum_{i=1}^{k} \alpha_t(i) \cdot \beta_t(i)} \tag{3.8}$$

$$\xi_t(i,j) = \frac{\alpha_t(i) \cdot a_{ij} \cdot b_j(\mathbf{x}^{(t+1)}) \cdot \beta_{t+1}(j)}{\sum_{p=1}^{k} \sum_{q=1}^{k} \alpha_t(p) \cdot a_{pq} \cdot b_q(\mathbf{x}^{(t+1)}) \cdot \beta_{t+1}(q)} \tag{3.9}$$

where a_{ij} denotes the transition probability from state s_i to state s_j, and $b_j(\mathbf{x}^{(t+1)})$ denotes the emission probability of $\mathbf{x}^{(t+1)}$ according to state s_j. The forward and backward variables are defined as follows:

$$\alpha_t(j) \stackrel{\text{def}}{=} P(S^{(t)} = s_j, \mathbf{x}^{(0:t)} \colon \lambda^{\text{old}}) \tag{3.10}$$

$$= \left[\sum_{i=1}^{k} \alpha_{t-1}(i) \cdot a_{ij} \right] b_j(\mathbf{x}^{(t)}) \tag{3.11}$$

$$\beta_t(i) \stackrel{\text{def}}{=} P(\mathbf{x}^{(t+1:T)} \mid S^{(t)} = s_i \colon \lambda^{\text{old}}) \tag{3.12}$$

$$= \sum_{j=1}^{k} a_{ij} \cdot b_j(\mathbf{x}^{(t+1)}) \cdot \beta_{t+1}(j) \tag{3.13}$$

where the basis of recursion is defined as $\alpha_0(i) = v_i b_i(\mathbf{x}^{(0)})$ and $\beta_T(i) = 1$ for all $i = 1, \ldots, k$, where v_i denotes the initial probability of state s_i. The variables α and β can be computed efficiently by means of dynamic programming, as illustrated by Proposition 3.3.

Proposition 3.3. *The computation of one E-step iteration for one sequence in asymmetric HMMs takes $\mathcal{O}(Tk^3n)$ time.*

It is straightforward to see that the cost of the E step in HMM-As is, in fact, the same as that of several other families of HMMs, including the independent and standard HMMs. We also note that the cost is strongly influenced by the number of states (which grows in a cubic fashion, whereas the other terms grow linearly).

3.4.3 *Maximization step*

In the M step, we obtain a new model λ^{new} based on the expected statistics previously computed. However, as opposed to the standard EM, the M step for HMM-As can no longer be computed efficiently in its exact form, as the graphical structure on the feature space is unknown, which relates to the intractable problem of finding the optimal structure of a Bayesian network (see Section 2.3.2). In fact, this efficiency can only be attained by very few families of HMMs, where the independent HMMs is the main one; even some models that do not capture asymmetries, e.g. the standard HMMs, also lose this property since the structure

is unknown (even though it is shared by all the states). To learn feature-space structures with reasonable quality, one often relies on approximate approaches.

In order to devise the update procedure for HMM-As, let us consider the expected score in SEM [14, 65]. The expected score for a candidate model λ is the expectation of the complete data likelihood taken with respect to the hidden states conditional on the current model λ^{old}:

$$\begin{aligned} Q(\lambda, \lambda^{\text{old}}) &= \mathbb{E}_{s^{(0:T)}}\left[\log P(\mathbf{x}^{(0:T)}, s^{(0:T)} \colon \lambda) - \text{Pen}(\lambda) \mid \mathbf{x}^{(0:T)} \colon \lambda^{\text{old}}\right] \\ &= \sum_{s^{(0:T)}} P(s^{(0:T)} \mid \mathbf{x}^{(0:T)} \colon \lambda^{\text{old}}) \cdot \log P(\mathbf{x}^{(0:T)}, s^{(0:T)} \colon \lambda) \\ &\quad - \text{Pen}(\lambda) \end{aligned} \tag{3.14}$$

The expectation is taken with respect to the latent state. Note that $P(\mathbf{x}^{(0:T)}, s^{(0:T)} \mid \lambda)$ factorizes according to the structure of the HMM-A (see Equation 3.5), thus:

$$\begin{aligned} Q(\lambda, \lambda^{\text{old}}) &= \sum_{s^{(0:T)}} P(s^{(0:T)} \mid \mathbf{x}^{(0:T)} \colon \lambda^{\text{old}}) \\ &\quad \cdot \log \left[P(s^{(0)}) \prod_{t=0}^{T-1} P(s^{(t+1)} \mid s^{(t)}) \prod_{t=0}^{T} P(\mathbf{x}^{(t)} \mid s^{(t)}) \right] \\ &\quad - \text{Pen}(\lambda) \end{aligned} \tag{3.15}$$

$$\begin{aligned} Q(\lambda, \lambda^{\text{old}}) &= \sum_{s^{(0:T)}} \log P(s^{(0)}) P(s^{(0:T)} \mid \mathbf{x}^{(0:T)} \colon \lambda^{\text{old}}) \\ &\quad + \sum_{s^{(0:T)}} \left(\sum_{t=0}^{T-1} \log P(s^{(t+1)} \mid s^{(t)}) \right) P(s^{(0:T)} \mid \mathbf{x}^{(0:T)} \colon \lambda^{\text{old}}) \\ &\quad + \sum_{s^{(0:T)}} \left(\sum_{t=0}^{T} \log P(\mathbf{x}^{(t)} \mid s^{(t)}) \right) P(s^{(0:T)} \mid \mathbf{x}^{(0:T)} \colon \lambda^{\text{old}}) \\ &\quad - \text{Pen}(\lambda) \end{aligned} \tag{3.16}$$

Equation 3.15 suggests that each term of the expected score can be optimized separately. The result is the parameter updating in the SEM process as follows.

3.4.3.1 *Structure learning*

In Equation 3.15, we identify the term associated to the emissions as:

$$\begin{aligned} &\sum_{s^{(0:T)}} \left(\sum_{t=0}^{T} \log P(\mathbf{x}^{(t)} \mid s^{(t)}) \right) \cdot P(s^{(0:T)} \mid \mathbf{x}^{(0:T)} \colon \lambda^{\text{old}}) - \text{Pen}(M_{\downarrow}) \\ &= \sum_{j=1}^{k} \sum_{t=0}^{T} \log P(\mathbf{x}^{(t)} \mid s_j^{(t)}) \cdot P(s_j^{(t)} \mid \mathbf{x}^{(0:T)} \colon \lambda^{\text{old}}) - \text{Pen}(M_{\downarrow}) \\ &= \sum_{j=1}^{k} \sum_{t=0}^{T} \gamma_t(j) \cdot \log P(\mathbf{x}^{(t)} \mid s_j^{(t)}) - \text{Pen}(M_{\downarrow}) \end{aligned} \tag{3.17}$$

The emissions term (Equation 3.17) can be further decomposed *per state*, because the state-specific networks are independent of each other. The advantage now is that *each state-specific network can be locally learned*. In this work, the penalty term is defined according to the BIC score (see Section 2.3.2). Thus, for a state s_j, its fraction from the emissions term is:

$$\begin{aligned}&\sum_{t=0}^{T} \gamma_t(j) \log P(\mathbf{x}^{(t)} \mid s_j^{(t)}) - \text{Pen}(M_{\downarrow}; s_j) \\ &= \sum_{t=0}^{T} \gamma_t(j) \log P(\mathbf{x}^{(t)} \mid s_j^{(t)}) - \frac{K_j \log(T+1)}{2}\end{aligned} \tag{3.18}$$

where K_j is the number of parameters in the model for state s_j. In HMM-As we wish to learn state-specific graphical structures in the M step, hence we run structure learning for each of the k states separately. In practice, structure learning often relies on approximate methods for exploring the search space of structures in feasible time and yet providing reasonable solutions.

In this work, we consider the tabu search [77] (TS, for short) to explore the candidate space of structures, which is a polynomial-time procedure based on hill-climbing search. A TS iteration explores the neighborhood of the current solution (initially an empty network) by adding, deleting or reversing an arc from this solution. The current solution is added to the tabu list, which stores the 10 most recently explored networks, in this implementation. Furthermore, only neighborhood solutions that are not in the tabu list are added to the neighborhood set (initially empty). Once the neighborhood set has been updated, the best of its solutions is taken out and set in the next iteration as the current solution. The new current solution might not be better than the previous one, however, this is allowed for no more than 10 consecutive iterations.

During the tabu search for the s_j state, the corresponding term in Equation 3.18 is used to compare candidate structures. Once the stopping criterion is reached in TS, the best structure that has been seen is returned. Stopping criteria include, e.g., testing whether the neighborhood set is empty, or testing if more than 10 iterations without improvement have passed. Given the described steps for TS, the cost of each structure learning run is bounded by a polynomial cost on the method's hyperparameters aforementioned and the number of observable features.

3.4.3.2 *Parameter update*

After obtaining a model structure for λ, it is possible to show that maximizing the expected score (Equation 3.15) leads to the following update formula for the transition probabilities:

$$\hat{a}_{ij} = \frac{\sum_{t=0}^{T-1} \xi_t(i,j)}{\sum_{t=0}^{T-1} \gamma_t(i)} \tag{3.19}$$

The update of the emission probabilities, in turn, is more involved than in standard EM, as the feature space is multivariate and each feature can have other

parents beyond the state variable. Furthermore, the parent set for a given feature can vary among states. Nevertheless, we can take advantage of the fact that the state-specific BNs allow us to factorize the joint distribution of the feature set $\mathbf{X}$, thus we can update the probability tables for one variable at a time. For state s_j and feature X, we update the corresponding probability tables as follows:

$$\hat{b}_j(X = x, \pi_j(X) = y) = \frac{\sum_{t=0}^{T} \gamma_t(j) \cdot \mathbb{1}(x^{(t)}, \pi_j(X)^{(t)} = y)}{\sum_{t=0}^{T} \gamma_t(j) \cdot \mathbb{1}(\pi_j(X)^{(t)} = y)} \tag{3.20}$$

where $\mathbb{1}$ is the indicator function. As in the case of arbitrary Bayesian networks, the cost of this calculation strongly depends on the connectivity of the network, being exponential in the number of features in the worst case. However, if the parent sets have moderate sizes, this can be very reasonable in practice.

As a final remark in learning HMM families, we note that a simpler version of this M step is needed for learning standard HMMs. In that case, structure learning is executed only once, as all the states will share a single structure. Analogously, updating the parameters in standard HMMs can still be costly due to the reasons previously discussed.

3.5 ASSESSMENT VIA SIMULATIONS

In this section, we aim to understand how unstructured, structured, and asymmetry-aware models cope with data generated from structured distributions. We also intend to analyze the effect of different amounts of data in model quality. To this end, we generated data from HMM-A distributions to simulate different scenarios. The model selection procedure used to learn models is described, as well as the data generation process, and finally the obtained results are discussed.

3.5.1 *Model selection*

In order to learn models that generalize best, we considered a model selection procedure to determine state spaces balancing complexity and overfitting avoidance as follows. Given a sequence dataset D, models are learned incrementally by increasing the number of states until overfitting occurs, which corresponds to the point where model score no longer increases. Model scoring is based on a 10-fold cross-validation: for each fold, a model is learned using training data (90% of the data) and its log-likelihood over validation data (the remaining data) is computed; after processing all the folds, the mean log-likelihood is taken, corresponding to the final score. To better assess learning, we learn 30 initial models for each k states, and select the one that generalizes the best to represent models with k states. Once the number of states has been determined, the final model L is learned using the entire dataset and those initial parameters, and it is evaluated by means of 60 independent datasets (not used in learning nor validation; each independent dataset has 2,000 sequences with length 20 each).

Each learned model L is evaluated by comparing likelihoods as follows. Let R be the true model used to sample D, then we define the fit quality of L as $\log \mathcal{L}_R - \log \mathcal{L}_L$, where $\log \mathcal{L}_R$ and $\log \mathcal{L}_L$ denote the mean log-likelihood of the models R and L over testing data respectively. This fit quality can be interpreted as the logarithm of the number of times the likelihood of the true model R is in comparison with the likelihood of the learned model L (in non-logarithmic scale). Hence, if the difference equals zero, it indicates that L and R fit equally well, while a difference larger than zero indicates that L fits worse than R. Thus, learned models with log-likelihood difference closer to 0 are preferred. We finally note that this procedure allows us to compare models learned with different amounts of data, as they are evaluated over the same testing datasets.

3.5.2 *Datasets*

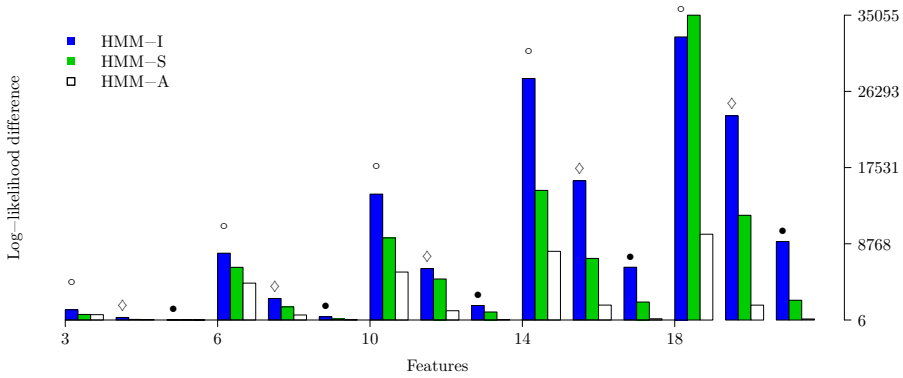
Datasets were sampled from random HMM-As, which were generated taking into account that many real-life networks have an average degree between 2 and 4 per node (i.e. the sum of in- and out-degrees). This is the case, for example, in well-known BNs, such as alarm, pathfinder, asia, and insurance [151]. Hence, in order to generate ground truth models having state-specific BNs with a reasonable, and yet realistic structure, the maximum degree of each node on each network was set to 3.

In order to build a random HMM-A with k states, its initial and transition matrices are sampled from Dirichlet distributions with concentration parameters all set to 1. Thus, valid distributions are obtained, i.e. matrices with rows that sum to 1 [69]. The emissions are Bayesian networks made of uniformly sampled DAGs [122, 151], whose nodes have the aforementioned maximum degree. All observables are modeled as random variables following Bernoulli distributions, whose parameters are sampled from Dirichlet distributions as before. We note that this procedure is also used to generate the initial models used in learning (see Section 3.5.1), except that no maximum degree is set. Finally, in the constructed scenarios the following quantities were considered: number of features $n \in \{3, 6, 10, 14, 18\}$, the state space dimension of true models $k \in \{2, 6\}$, and the amount of sequential data as 50 sequences (each with length of 10 time points), 200 sequences (10 time points) and 1,000 sequences (20 time points).

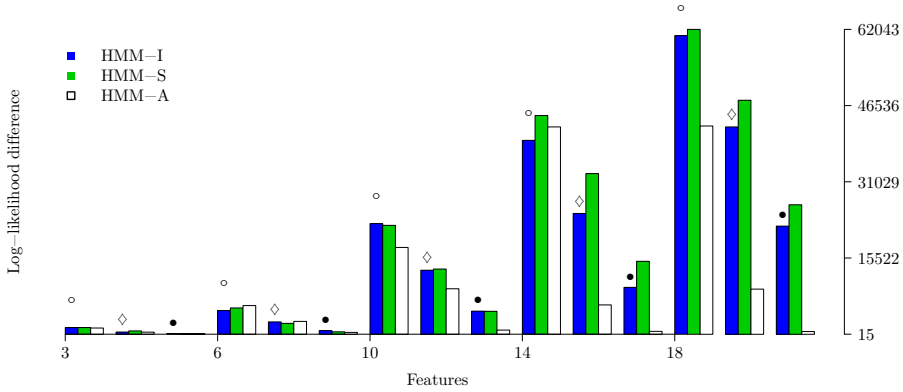
3.5.3 *Results for symmetric models*

Figure 3.4 shows the log-likelihood differences between asymmetric and symmetric HMMs based on simulated data (here, HMM-S refers to standard HMM). We first note that, as expected, all the classes of models obtained better fit when more data is provided, which is influenced by the fact that more states can be learned prior to overfitting. The results also suggest that independent and standard HMMs were not able to provide the same model quality as HMM-As, even when the highest amounts of data were provided to all the three models. Hence, it

seems reasonable to expect that much more data would be needed in order to learn models that fit as well as the learned HMM-As (in this case, using 1,000 sequences). Concerning the scarcer datasets (note that the larger datasets are 40× larger than the smaller ones), HMM-As achieved superior model quality on most cases. This allows us to conclude that HMM-As showed a good compromise in a varied range of dataset sizes, which can be explained by its flexibility on learning more or less dense feature-space structures depending on the situation.



(a) Number of states in true models = 2.



(b) Number of states in true models = 6.

Figure 3.4: Fit of asymmetric and symmetric HMMs learned in simulations. Datasets sampled from true models have 50 sequences (length 10 time points, ◦), 200 sequences (length 10 time points, ◊), and 1,000 sequences (length 20 time points, •). Note that scales on Y axes differ.

In terms of scaling, e.g. when modeling more observables, the additional structure of HMM-As avoided pitfalls that can hinder independent and standard HMMs: HMM-Is will tend to increase their state space, while standard HMMs will tend to model denser feature-space graphical structures. As a consequence, in most cases these symmetric models approach overfitting with much less model

quality than HMM-As. In other words, despite the representational equivalence between HMM-As and independent and standard HMMs in theory, such symmetric models can be limited in practice. These claims are further supported by results in Table 3.2 showing the corresponding state space dimensions, and Figures 3.5a-3.5b showing the number of parameters.

Table 3.2 shows the dimension of state spaces associated to learned models, suggesting that approximating HMM-A distributions required independent HMM with state spaces substantially larger than the true models' spaces, while this was not the case for standard HMMs. Nevertheless, as Figures 3.5a-3.5b show, the number of parameters in these two families were substantially higher than those of learned HMM-As, specially when more features were involved. With regard to running time in learning, Figures 3.6a and 3.6b show that, somewhat surprisingly, learning HMM-Is was more costly in most cases than HMM-As: although learning HMM-As is done via structural EM, its combination with search heuristics and smaller space state was in practice more efficient than the EM used to learn HMM-Is.

n	**HMM-I**	**HMM-S**	**HMM-A**	n	**HMM-I**	**HMM-S**	**HMM-A**
3	3	2	2	3	3	3	3
6	3	2	2	6	3	2	2
10	6	2	2	10	6	2	3
14	5	2	2	14	7	2	4
18	5	2	2	18	7	2	5
Number of states in true models = 2				Number of states in true models = 6			

(a) **Dataset size = 50 sequences**.

n	**HMM-I**	**HMM-S**	**HMM-A**	n	**HMM-I**	**HMM-S**	**HMM-A**
3	4	2	2	3	4	4	4
6	7	3	2	6	6	3	3
10	10	2	2	10	11	3	8
14	9	2	2	14	13	2	6
18	13	2	2	18	15	3	6
Number of states in true models = 2				Number of states in true models = 6			

(b) **Dataset size = 200 sequences**.

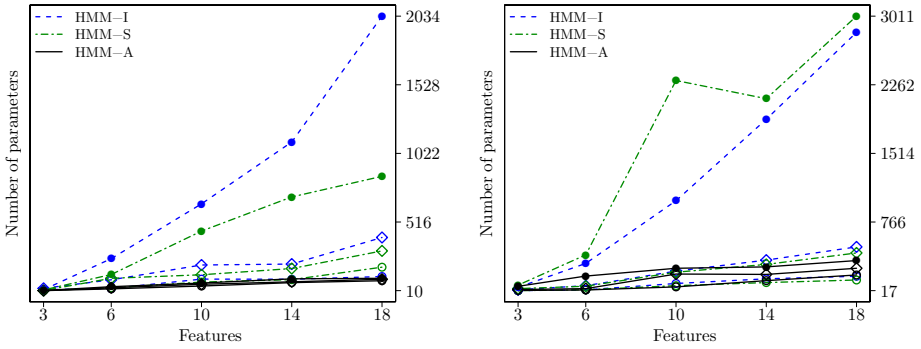
n	**HMM-I**	**HMM-S**	**HMM-A**	n	**HMM-I**	**HMM-S**	**HMM-A**
3	3	2	2	3	6	6	6
6	13	2	2	6	15	6	7
10	21	2	2	10	27	6	7
14	27	2	2	14	37	3	6
18	37	2	2	18	45	3	6
Number of states in true models = 2				Number of states in true models = 6			

(c) **Dataset size = 1,000 sequences**.

Table 3.2: State spaces of asymmetric and symmetric HMMs learned in simulations.

3.5.4 *Results for asymmetric models*

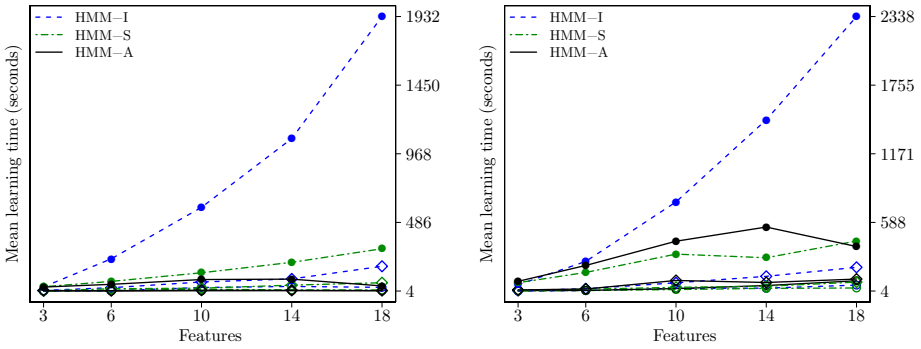
Figure 3.7 shows the fit quality results for HMM-As and Chow-Liu HMMs (HMM-CLs). These results indicate that restricting the feature space to trees



(a) Number of states in true models = 2.

(b) Number of states in true models = 6.

Figure 3.5: HMM-As and symmetric HMMs learned from simulated data: number of parameters for different cases.

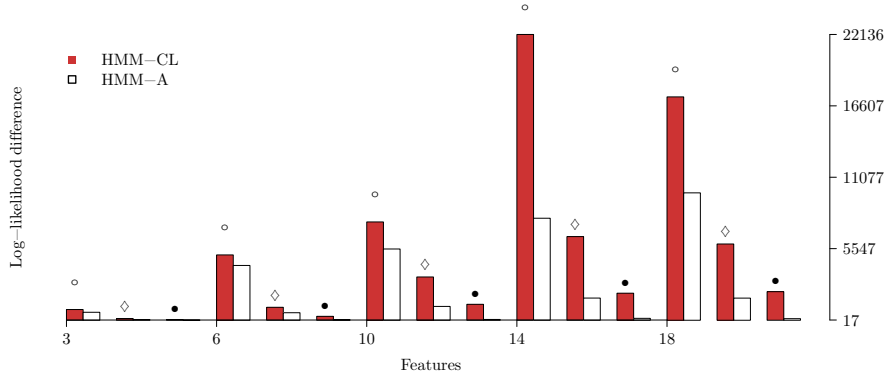


(a) Number of states in true models = 2.

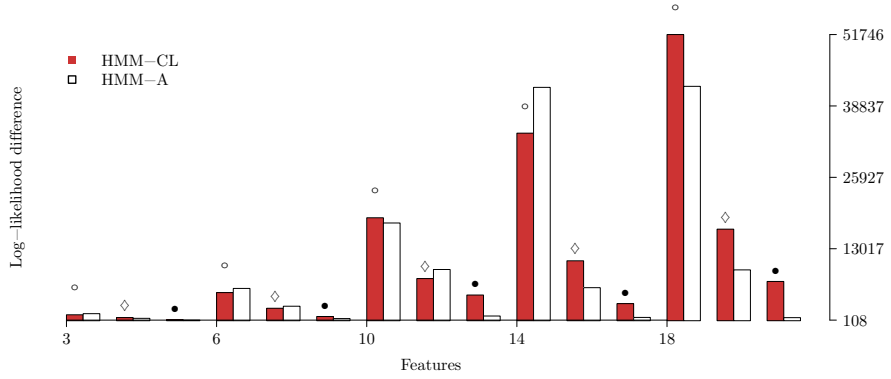
(b) Number of states in true models = 6.

Figure 3.6: HMM-As and symmetric HMMs learned from simulated data: mean learning time in seconds.

prevented HMM-CLs from achieving model quality as high as that by HMM-As on most of the considered scenarios. This is more evident in the cases involving more observables, where the learned HMM-As reached the most superior model quality compared to HMM-CLs, which is likely influenced by the size of the possible graphical structures for emissions, a situation which HMM-As can better handle since HMM-As are not restricted to trees. On the other hand, HMM-CLs are prone to be more efficient in practice, since learning Chow-Liu trees can be done efficiently per EM iteration [102]. Similarly to the symmetric models case, HMM-As could be trained with less data, and yet provided similar or better model quality than HMM-CLs – although here to a lesser extent when the data generating process had a higher number of hidden states. Furthermore, extending the state spaces of HMM-CLs resulted in better models, however, prior to overfitting these achieved lower quality than HMM-As.



(a) Number of states in true models = 2.



(b) Number of states in true models = 6.

Figure 3.7: Fit of asymmetric and Chow-Liu HMMs learned in simulations. Datasets sampled from true models have 50 sequences (length 10 time points, ◦), 200 sequences (length 10 time points, ◊), and 1,000 sequences (length 20 time points, •). Note that scales on Y axes differ.

A comparison based on Figures 3.4 and 3.7 suggests that modeling state-specific structures, whether by means of general asymmetries as HMM-As do or tree-shaped ones as HMM-CLs do, led to better results than those of symmetric models. HMM-As needed in general fewer states or fewer parameters than symmetric HMMs, which also holds for HMM-CLs with respect to symmetric HMMs, as shown in Table 3.3 and Figure 3.8. Hence, the results of this section suggest a somewhat consistent conclusion: capturing the distribution underlying data generated by more structured processes is more adequate by means of models that capture distribution specificities associated to the hidden states. Although symmetric models can in theory capture such distributions, whether by an increase of their state spaces or by modeling denser emissions structure, in many realistic situations – where data is often limited – the asymmetric models

exhibited several advantages and better handled the complexity *versus* quality trade-off.

n	HMM-CL	HMM-A	n	HMM-CL	HMM-A
3	2	2	3	3	3
6	2	2	6	2	2
10	2	2	10	3	3
14	3	2	14	4	4
18	2	2	18	4	5
k in true models = 2			k in true models = 6		

(a) **Dataset size = 50 sequences**.

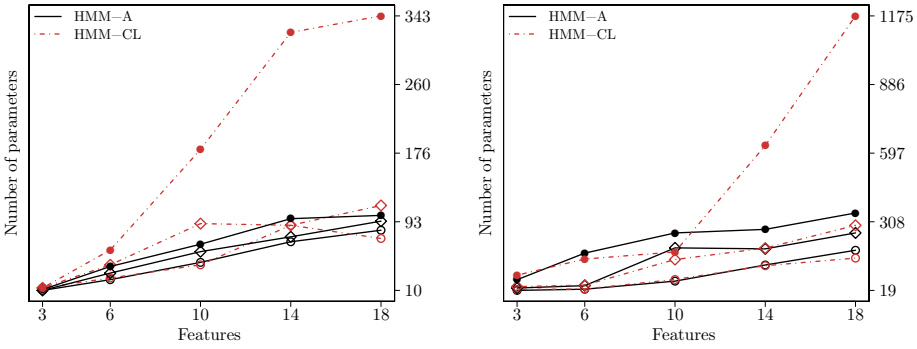
n	HMM-CL	HMM-A	n	HMM-CL	HMM-A
3	2	2	3	4	4
6	3	2	6	3	3
10	4	2	10	6	8
14	3	2	14	6	6
18	3	2	18	7	6
k in true models = 2			k in true models = 6		

(b) **Dataset size = 200 sequences**.

n	HMM-CL	HMM-A	n	HMM-CL	HMM-A
3	2	2	3	7	6
6	4	2	6	8	7
10	7	2	10	7	7
14	9	2	14	15	6
18	8	2	18	21	6
k in true models = 2			k in true models = 6		

(c) **Dataset size = 1,000 sequences**.

Table 3.3: State spaces of HMM-As and HMM-CLs learned in simulations (k denotes number of states).



(a) Number of states in true models = 2.

(b) Number of states in true models = 6.

Figure 3.8: HMM-As and HMM-CLs learned from simulated data: number of parameters for different cases.

3.6 EXPERIMENTS WITH REAL-WORLD DATASETS

In this section, we describe experiments for learning symmetric HMMs, HMM-CLs and HMM-As from real-world datasets originated from several domains. In order to empirically determine state spaces and assess the learned models, we used a procedure similar to the one described in Section 3.5. It differs in that real-life datasets are split in two parts: one for selecting models via 10-fold cross-validation (using 80% of the data), and the remaining portion for assessment of generalization.

3.6.1 *Datasets*

The datasets considered in this section are summarized in Table 3.4 and described next.

Dataset	n	**Description**	**Sequence data**
Volvo	3	Event logs of software incidents	151 (50)
Rabobank	6	Event logs of software incidents	500 (30)
Airquality	12	Urban pollution monitoring	40 (48)
Printer $\mathcal{R}_1$	7	Performance of printing nozzles and maintenance activity	27 (15)
Printer $\mathcal{R}_2$	7	Performance of printing nozzles and maintenance activity	52 (15)
Printer $\mathcal{R}_3$	7	Performance of printing nozzles and maintenance activity	58 (15)

Table 3.4: Summary of real-world datasets. The *sequence data* column shows the number of sequences together with sequence duration in parenthesis.

3.6.1.1 *Business process data*

The business process dataset consists of event-log records on software incidents related to, e.g., software bugs, hardware problems, among others within the scope of ICT company departments. In general, these datasets are often used for process mining, covering tasks such as conformance checking (i.e. checking whether the business process specification complies with the running process), process discovery and process enhancement [1]. Learning business models as done in process mining field often intends to capture the underlying sequential behavior of actions within events. Thus, given a collection of events, business models are fitted to this data in order to represent different ways in which an event can develop over its lifetime.

As opposed to business models (e.g. workflow-like models), where one often wants to understand the internals of events, in this section we learn a complementary behavior from event-logs data in the form of influences among events. As this is less evident from data and involves multivariate observations (since events

are typically composed of several features), this is a challenging problem, for which this section offers an HMM-based solution. We considered two datasets from the BPI (business process intelligence) challenge, described as follows.

VOLVO DATASET The Volvo IT Belgium dataset [166] consists of event logs of software incident registered during the period of 2011-2012. Each data point describes an incident by means of three features: incident impact, push to front (i.e. whether the incident was handled by a service desk team or required other specialized teams as well), and country (referring to whether the incident involved employees from different nationalities). The Volvo data was split in sequences such that each sequence has approximately 5 days of incidents.

RABOBANK DATASET The Rabobank Group ICT dataset [55] consists of event-log records of software incidents over the period of 2011-2014, however from a different software domain than the Volvo dataset. We considered the part of the data related to interactions, which registers the first contact between a user of a software component and a service desk team. An interaction call can lead to an incident or not. Each interaction is described in the Rabobank dataset by a set of six features: type of involved item (e.g. application, hardware, network-related issues), impact (in case of service disruption), priority, category (i.e. whether the event refers to a request for information or an incident), first call (i.e. whether the interaction could be solved by service desk team or led to an incident for further resolution), and handle time (i.e. the amount of time to resolve the service disruption).

Learning HMM-As for business processes aims in first place to provide well fitted models, but also aims to discover different dynamics that might govern the generation of incidents and interactions. This can then be turned into practical knowledge, e.g., to assist decision makers when devising more effective and resource-saving business processes. We shall discuss more on this in Section 3.6.3.

3.6.1.2 *Airquality data*

The Airquality dataset contains data on gas pollutants in the context of urban pollution monitoring [172]. The feature set is composed by two different sources of information: a set of reference pollutant concentrations provided by conventional stations, and a set of measurements provided by a multi-sensor device. Originally, the Airquality dataset was used to evaluate and calibrate sensor devices for estimating the concentration of pollutants, as a technological means for low cost and convenient air monitoring across urban spaces. In the original paper [172], simple positive correlations among sensors data were found to influence the prediction accuracy, hence we provide a complementary analysis to how these correlations develop in a sequential way. We considered a feature set with 12 variables corresponding to the original measurements, which were discretized for the experiments in this section. The records for the variable for the ground-truth

non-methane hydrocarbons were not considered, as they were absent in most of the cases.

3.6.1.3 *Printers data*

We also considered data to support understanding the behavior of modern, complex engineered artifacts (also called cyber-physical systems), for which we use a large-scale printer as a case study. Whereas engineers understand the functioning of the individual components in considerable depth and detail, as a consequence of their intricate design they find it much more difficult to understand the behavior of the artifacts at a certain level of abstraction, as well as their interaction. In order to learn the temporal behavior of such systems, data was gathered from three printers of the same printer family, where the usage of the printers differs as function of time, and as function of the print jobs being rendered. In this case study, we focus in particular on one component – the nozzle – that aims at jetting ink on the paper. The behavior of nozzles as function of time depends on several factors, such as the quantity of ink used, time since last maintenance and some environmental parameters.

The logs that were considered consist of a 1-year record of nozzle-related factors continuously monitored. We considered a key maintenance action that is performed by the machine from time to time, and gathered data on nozzle-related components between each maintenance occurrence, such that each (multivariate) observation includes the following features: interval duration (i.e. the length of time since the previous maintenance action), total workload, frequency of another related maintenance action, and color-related features. The goal of our experiment is to discover relations between features and how it influences the proper functioning of the nozzles.

3.6.2 *Results*

We first report results on fit quality based on model selection, where Figure 3.9 shows the mean validation log-likelihoods in function of the number of states. These results show that the structural simplicity of independent HMMs could be compensated to some extent by learning larger state spaces, and thus model quality similar to that attained by more structured models (i.e. standard HMMs and HMM-As) could be achieved. However, this was not possible in all the cases, in particular in the business process datasets. In these cases, prior to achieving overfitting the structured models had a much better fit, suggesting that in some cases the presence of non-trivial structure over observables can be deemed crucial in order to obtain good models.

With respect to the structured models, contrasting standard HMMs with HMM-As indicates that HMM-As achieved superior fit on some cases (e.g. Airquality and Rabobank) and similar model quality on the remaining ones. The learned HMM-As better fitted the data than Chow-Liu HMMs in general as well. It is interesting to note that HMM-CLs impose tree structures to its emissions,

something that might not be always beneficial. For example, in the Volvo and Airquality cases, the results suggest that even a symmetric model as the HMM-S was able to provide better results than HMM-CLs, which is interesting as the HMM-S does not necessarily learn a connected structure for the emissions space. In general, it can also be observed from Figure 3.9 that structured models as standard HMMs and Chow-Liu HMMs overfit much more easily than HMM-As, suggesting that HMM-As provide more parsimonious solutions to these real problems.

As Figure 3.9 shows, dynamic Bayesian networks (DBNs) were also learned from the real-world datasets, whose results indicate that DBNs provided consistently inferior model quality than HMMs. Although these results are not directly related to comparing HMMs, they suggest that modeling autoregressions alone (as in DBNs) is not a guarantee for good fit in real-life datasets: modeling multiple (and possibly structured) distributions via hidden states can be more powerful, yet no autoregressions are modeled by these HMMs. A question that could be of interest is whether including autoregressions in HMM-As would bring real benefits to such models.

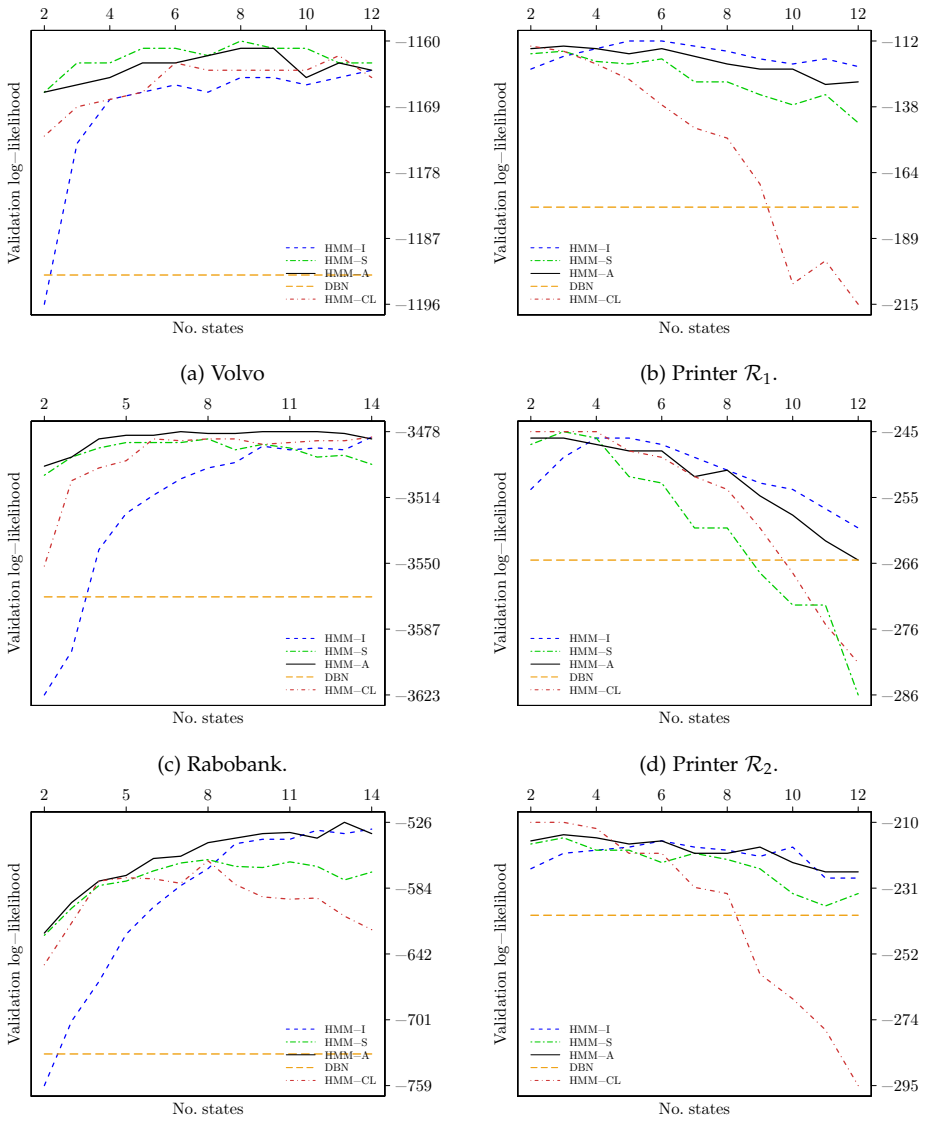
Having discussed the dynamics of model quality based on validation log-likelihoods experiments, we now use these results to select and learn models in order to discuss problem insight, as well as to assess their generalization. To this end, we select models in a flexible way: we pursue models with the highest fit, except when there are multiple models with similar quality, in which case we select the models with the lowest dimension. After finishing this, we learn models with the selected dimension using the entire datasets and measure their likelihood with testing data (i.e. data that was not used in cross-validation).

The models learned for generalization assessment are summarized in Table 3.5. As there is no ground-truth model for the real-world datasets, to facilitate comparison we used *normalized log-likelihoods* as follows:

$$\mathrm{NLL} = \frac{-\log \mathcal{L}(\mathcal{B})}{c} \tag{3.21}$$

where $\log \mathcal{L}(\mathcal{B})$ is the log-likelihood of the model $\mathcal{B}$, and c is a normalizing constant given by $c = mTn$, with m being the number of sequences, T the sequence length, and n the number of features in the dataset.

As Table 3.5 shows, HMM-As generalized consistently better than symmetric HMMs and Chow-Liu HMMs. We further computed 95% confidence intervals (CIs) for these models as shown in Table 3.6, in order to check for the robustness of the generalization assessment (intervals were obtained by means of bootstrapping the testing datasets for 2,000 times in each case). The CIs show that HMM-As could provide significantly better model quality on most scenarios of business process and Airquality cases. Significance in favor of HMM-As was also obtained in the printers cases, although HMM-As can be considered better than Chow-Liu HMMs in these cases but not significantly. This is explained by the fact that the HMM-A states are all virtually associated to forests that are sparser than trees, as in Printer $\mathcal{R}_3$ (Figure 3.14) and Printers $\mathcal{R}_1$ and $\mathcal{R}_2$ as well [23].



(a) Volvo

(b) Printer $\mathcal{R}_1$.

(c) Rabobank.

(d) Printer $\mathcal{R}_2$.

(e) Airquality.

(f) Printer $\mathcal{R}_3$.

Figure 3.9: Cross-validation log-likelihoods achieved by DBNs, symmetric, Chow-Liu, and asymmetric HMMs in real-world datasets. Each point represents the mean validation log-likelihood over 10 folds.

From the results on real-data discussed in this section, it seems fair to conclude that not only more structure is beneficial for HMMs to better capture real-life problems, but also the *right* additional structure as provided by HMM-As by their state-specific Bayesian-network distributions. The number of parameters in HMM-As was consistently lower than those of standard HMMs, independent

	HMM-I			HMM-S			HMM-CL			HMM-A		
Dataset	k	**NLL**	**#Pa.**	k	**NLL**	**#Pa.**	k	**NLL**	**#Pa.**	k	**NLL**	**#Pa.**
Volvo	8	65.2	87	5	**64.7***	59	6	65.2	65	5	**64.7***	50
Rabobank	10	**48.5***	159	5	48.7	214	6	**48.5***	101	4	**48.5***	73
Airquality	8	32.9	159	8	32.1	623	8	35.4	247	6	**31.3***	170
Printer $\mathcal{R}_1$	5	49.8	59	3	48.7	53	3	46.6	47	3	**46.0***	36
Printer $\mathcal{R}_2$	4	60.5	43	3	61.7	65	3	60.4	47	3	**59.9***	43
Printer $\mathcal{R}_3$	4	48.0	43	3	46.0	56	3	46.4	47	3	**45.7***	36

Table 3.5: Generalization assessment of learned models on real-world datasets. Notation: k denotes the number of states, NLL the normalized log-likelihood, and *#Pa.* the number of parameters. Results that generalized the best are bold-faced and followed by an asterisk.

Dataset	**Asymmetric vs. Independent**	**Asymmetric vs. Standard**	**Asymmetric vs. Chow-Liu**
Volvo	**[-0.86, -0.11]****	$[-0.14, 0.29]\dagger_{\mathrm{S}}$	**[-0.78, -0.21]****
Rabobank	$[-0.15, 0.15]$	$[-0.47, 0.06]*_{\mathrm{A}}$	$[-0.02, 0.26]\dagger_{\mathrm{CL}}$
Airquality	**[-3.17, -0.17]****	$[-3.08, 0.61]*_{\mathrm{A}}$	**[-10.30, -1.77]****
Printer $\mathcal{R}_1$	$[-10.35, 0.46]*_{\mathrm{A}}$	$[-7.88, 0.37]*_{\mathrm{A}}$	$[-1.32, 0.05]*_{\mathrm{A}}$
Printer $\mathcal{R}_2$	$[-1.53, 0.74]*_{\mathrm{A}}$	**[-4.16, -0.38]****	$[-1.58, 1.11]*_{\mathrm{A}}$
Printer $\mathcal{R}_3$	**[-4.79, -0.3]****	$[-1.67, 0.26]*_{\mathrm{A}}$	$[-3.35, 3.48]$

Table 3.6: 95% bootstrap confidence intervals for the differences on generalization assessment (real-world datasets). Negative values indicate better fits for HMM-As. Notation: $**$ = HMM-A is significantly better; $*_{\mathrm{A}}$ = HMM-A is better but not to a significant extent; $\dagger_{\mathrm{X}}$ = model X is better but not to a significant extent.

HMMs and Chow-Liu HMMs, suggesting that diverse local structure exists which could be discovered by HMM-As.

3.6.3 *Problem insight*

We discuss in this section problem insight that can be gained from the learned HMM-As. We stress that from a fundamental perspective, where Bayesian networks are tools to facilitate reasoning with statistical independences, the fact that HMM-As can provide multiple graphical structures to explain how dynamic systems evolve over time (e.g. a business process) represents additional insight by its very nature. This contrasts to symmetric HMMs, where all those specificities are lost (or hidden across a number of CPTs at most), thus much less insight is likely to be gained.

3.6.3.1 *Business process models*

Figure 3.10 shows the HMM-A learned from the Rabobank case (CPTs are not shown). The model shows that *Type* is unconditionally independent of *Impact* and *Priority* on the bottom right-most state, while this is not the case on the top right-most state. This structural information might be used, e.g., to further develop different policies for scheduling different types of interactions in different moments: if the system is assumed to be in the bottom right-most state, a more flexible scheduling might be possible, where different types of interactions do not need to be handled by priority or impact, but instead could be handled by the expected time to be solved (due to the relationship with *Handletime*). On the other hand, if the system is in the bottom left-most state, *Type* is still unconditionally independent of *Impact*, but its unconditional independence of *Priority* no longer holds: in fact, such state seems to act as a bridge for the two aforementioned states.

The aforementioned problem insight cannot be derived from the (almost fully connected) graphical structure of the learned HMM-S partially shown in Figure 3.11, nor from the learned HMM-I. At a higher level of abstraction, HMM-As also allow for new insight obtained by combining the local state properties with state-transition probabilities: this shows that batches made of few software-incident events that share independence properties are produced over time.

Figure 3.12 shows the HMM-A learned from the Volvo dataset. As in the Rabobank case, this HMM-A is made of different graphical structures that lead to different independence relations, whereas the standard HMM has a fully connected structure as shown in Figure 3.13. Finally, we note that in asymmetric models, not only probabilistic relationships change, but also the structure in each state, providing evidence that these models capture an additional facet of the different stages the underlying dynamic system can transit to.

3.6.3.2 *Printers model*

Figure 3.14 shows the HMM-A learned from Printer $\mathcal{R}_3$ dataset (the other printer models were discussed elsewhere [23]). This model suggests that the behavior of such large-scale printer alternates between two modes in the long run, which can be distinguished based on how the color rates C_1, C_2, C_3, and C_4 interact with the other observables. For example, once the printer is assumed to be in the right-most top state, one could decide on whether the number of maintenance performed could be altered in order to save resources, as this variable will not affect the colors' performance. However, this is probably not the case for most of the colors if the printer is in the center state, where those colors do interact with other observables. The standard HMM learned for this printer is shown in Figure 3.15, lacking from such specific alternation behavior that could be discovered by means of the HMM-A, as the colors variables are connected to all the other observables (whether directly or not).

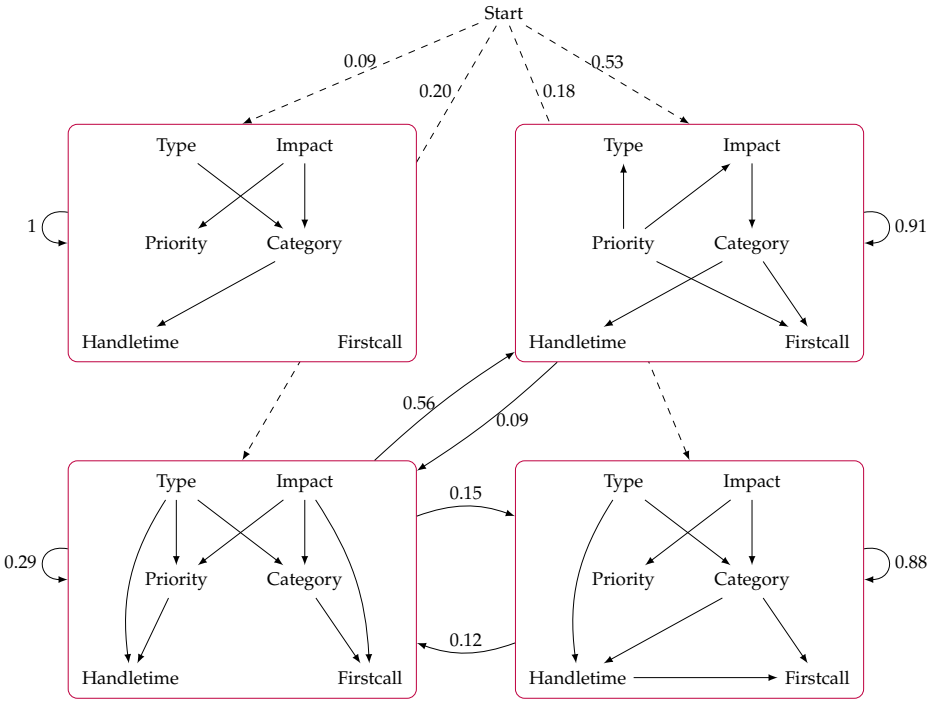


Figure 3.10: HMM-A learned from the Rabobank dataset. Dashed arcs indicate initial probabilities.

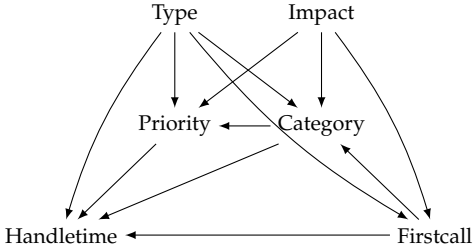


Figure 3.11: Graphical structure for emissions of the standard HMM learned from Rabobank dataset.

3.7 RELATED WORK

Analyses of the sensitivity of Bayesian networks to parameter change are relatively numerous [34, 72, 130], however that does not seem to be the case when it comes to the sensitivity to the graphical structure. There is some research on how model structure affects accuracy in medical diagnosis problems [131], where the authors have shown that the accuracy was not significantly sensitive for disturbances on model structure, considering certain medical cases and diagnostic criteria. In the

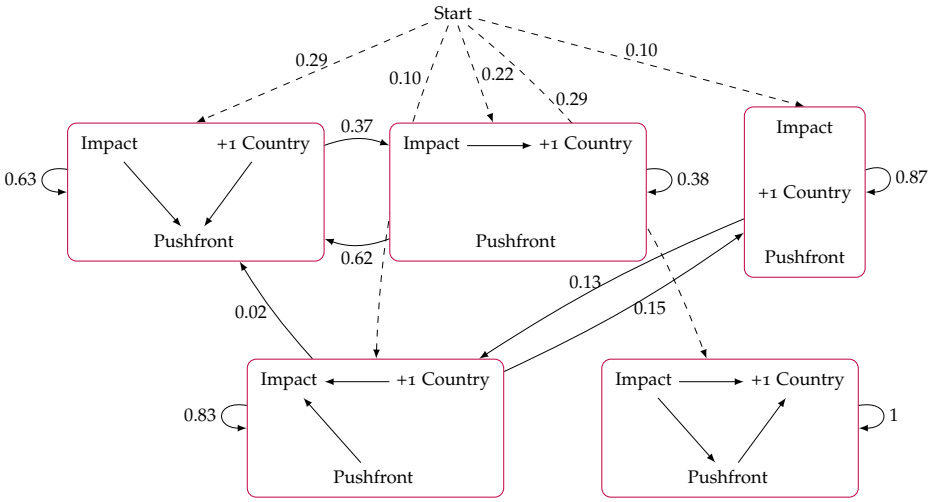


Figure 3.12: HMM-A learned from Volvo dataset.

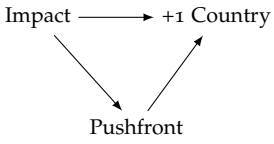


Figure 3.13: Graphical structure for emissions of the standard HMM learned from Volvo dataset.

context of HMMs, however, the results shown in this chapter suggest a different conclusion: with respect to model fit, modeling additional and specific (by means of distribution asymmetries) seemed very important for achieving better model quality. Nevertheless, these conclusions are not necessarily contradictory in principle, as the employed criteria differ and so do the type of models. As in HMMs the state space dimension is a rather important parameter, modeling non-trivial structure that can lead to smaller state space seems crucial to such models, while this might not be the case for some static Bayesian networks. In fact, it is a general belief in the Bayesian networks field that non-trivial structure matters for better handling real-world problems [51, 67, 150].

In this work, we attempted to provide a better understanding of the effects of modeling asymmetries on the feature space in HMMs, which is somewhat lacking in the literature. This involved a more thorough comparison of HMM-As with several families of HMMs: this included not only independent HMMs, but also standard HMMs and Chow-Liu HMMs, the latter being able to model simpler asymmetries than HMM-As. Finally, the chapter showed experiments involving models of DBNs, which are typically learned without hidden or latent variables.

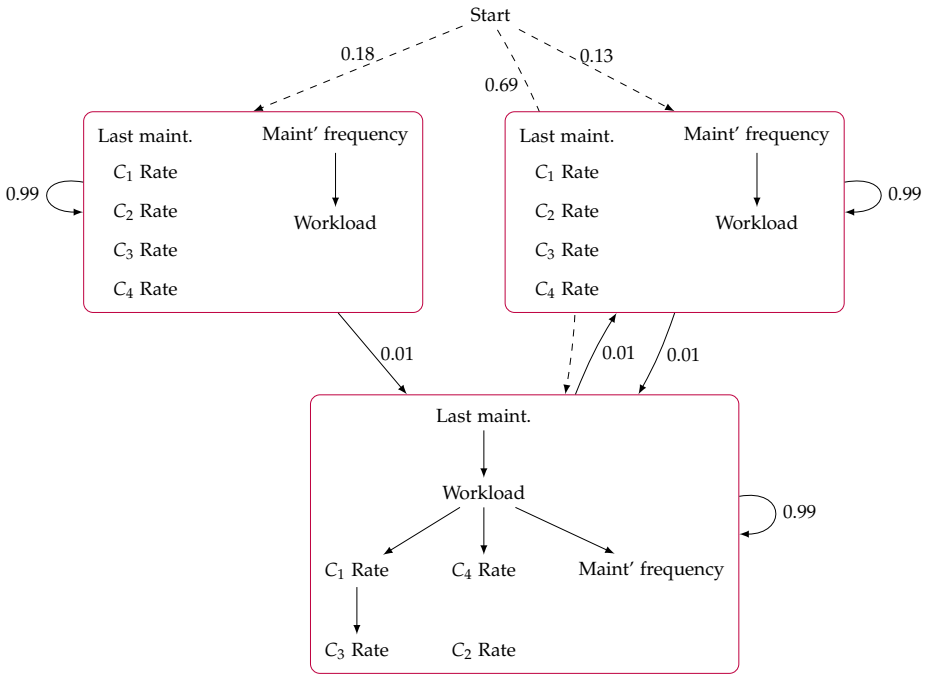


Figure 3.14: HMM-A learned from Printer $\mathcal{R}_3$ case.

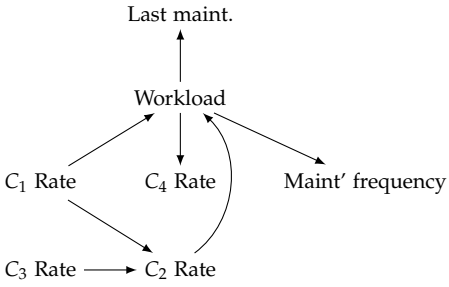


Figure 3.15: Graphical structure for emissions of the standard HMM learned from Printer $\mathcal{R}_3$ dataset.

3.8 CONCLUSIONS

In this chapter, we proposed a new family of HMMs called asymmetric hidden Markov models. HMM-As explicitly capture distribution asymmetries inherent to many real-world problems, by means of associating individual hidden states to arbitrary Bayesian networks. We showed that, in principle, symmetric HMMs (e.g. independent and standard HMMs) can have their state space or emissions structure arbitrarily extended for representing structured distributions.

Nevertheless, empirical results showed that this capability was not enough for guaranteeing comparable model quality, due to model overfitting (because of too many states or too dense emission structures). A similar conclusion holds for Chow-Liu HMMs, suggesting that going beyond tree-shaped asymmetries as done by HMM-As can be beneficial.

In some real-world cases, adding structure, either via symmetric or asymmetric models, allowed for relevant model quality improvement, while the simplest model (i.e. the independent HMM) was good enough in other cases. This model selection issue could be adequately addressed by HMM-As, which provide enough flexibility to reduce the need for selecting a particular HMM architecture a priori.

Computationally, learning HMM-As introduces an additional burden due to structure learning, compared to symmetric HMMs. Nevertheless, experiments indicated that good-quality HMM-As with compact state spaces could be obtained by using graphical structures found by common search heuristics. Hence, in practice learning HMM-As using structural EM resulted in fact in shorter running times compared to learning symmetric HMMs using standard EM in many cases. Furthermore, HMM-As learned from real-world datasets with varied sizes and number of observables were shown to bring additional problem insight that cannot be readily obtained from symmetric HMMs.

Several paths for further research can be considered. To some extent, HMM-As can be seen as tools for summarizing hidden Markov models with larger states spaces into models with more compact state spaces, as shown in the real-world experiments. We would like to further evaluate whether HMM-As act as *model summarizers* in more general settings, e.g. when the data generation mechanism has no explicit asymmetries (as in HMM-Is), and when it consists of different kinds of asymmetries (such as autoregressions, as in dynamic multinets).

It could be also of interest to exploit the sensitivity of differences between state-specific networks, e.g., along the lines of sensitivity analysis research. This could help, e.g., to eliminate too specific arcs that do not significantly contribute to model quality, thus allowing for more compact models. As we observed in real-world experiments and in simulations, several advantages obtained with HMM-As were more prominent when the problem had higher number of observables. Hence, we intend to further investigate such scaling aspect, as well as consider other real-world cases with more features and different types of observables (e.g. continuous and hybrid ones). Finally, we would like to compare the identification of asymmetries in sequential models as HMM-As with other approaches, such as knowledge compilation [39] and dynamic chain event graphs [5].

3.A PROOFS

Proof of Proposition 3.1. Let $\mathcal{M}$ be the given HMM-A and $\mathcal{M}'$ be a standard HMM, where both models are defined over $(\mathbf{X}, S)$. We construct $\mathcal{M}'$ for simulating $\mathcal{M}$ as follows. Let G_F be directed acyclic graph over $\mathbf{X}$ that is also fully connected. Add an arc from S to each $X_i \in \mathbf{X}$, and define the result as the graphical structure of

the emissions of $\mathcal{M}'$. By the chain rule from probability theory, a fully connected structure can represent any probability distribution, hence, the distribution of each state in $\mathcal{M}$ can be represented by a state in $\mathcal{M}'$ by adequately parameterizing the CPTs on the emissions of $\mathcal{M}'$. This allows us to obtain $P'(\mathbf{X}^{(t)} \mid s^{(t)}) = P(\mathbf{X}^{(t)} \mid s^{(t)})$, for every state $s \in \mathrm{dom}(S)$.

Finally, we set the initial and transition distributions of $\mathcal{M}'$ to the same as those from $\mathcal{M}$. Thus, we conclude that $P'(\mathbf{X}^{(0:T)}, S^{(0:T)}) = P(\mathbf{X}^{(0:T)}, S^{(0:T)})$. □

Proof of Proposition 3.2. We construct in the following an independent HMM $\mathcal{M}'$ for simulating a given asymmetric HMM $\mathcal{M}$ with k states. The observables are assumed to follow Bernoulli distributions each (an extension to multinomial distribution is straightforward). We denote by P and P' the joint distributions over $(\mathbf{X}^{(0:T)}, S^{(0:T)})$ of $\mathcal{M}$ and $\mathcal{M}'$ respectively. Note that the state-specific BN associated to any state in $\mathcal{M}$ is a BN over n variables, hence its joint distribution can be completely characterized with at most $2^n - 1$ independent parameters, where we denote by $\theta_{\mathbf{x}}$ the parameter associated to the assignment $\mathbf{x}$. For each $\theta_{\mathbf{x}}$ from state s_i, we define a state $s_{i\theta_{\mathbf{x}}}$ in $\mathcal{M}'$, as well as emission distributions of the form $P'(X_i = \top \mid s_{i\theta_{\mathbf{x}}}) \stackrel{\text{def}}{=} 1$ whenever $(X_i = \top)$ holds in $\mathbf{x}$. Following this procedure for all state-specific BNs from all states will result in $k2^n$ states in total in $\mathcal{M}'$. This finishes the construction of the emission distribution for $\mathcal{M}'$.

The remaining distributions of $\mathcal{M}'$ are constructed by scaling the corresponding distributions in $\mathcal{M}$ with the probability of each joint assignment of $\mathbf{X}$ as follows. For the initial distribution, we define

$$P'(s^{(0)}_{i\theta_{\mathbf{x}}}) \stackrel{\text{def}}{=} P(s^{(0)}_i) P(\mathbf{x}^{(t)} \mid s^{(t)}_i)$$

for each state s_i from $\mathcal{M}$ and assignment $\mathbf{x}$. On the other hand, we define the transitions in $\mathcal{M}'$ as

$$P'(s^{(t+1)}_{j\theta_{\mathbf{x}}} \mid s^{(t)}_{i\theta_{\mathbf{x}}}) \stackrel{\text{def}}{=} P(s^{(t+1)}_j \mid s^{(t)}_i) P(\mathbf{x}^{(t+1)} \mid s^{(t+1)}_j)$$

where $s_{i\theta_{\mathbf{X}}}$ refers to any state originated from s_i. Here the instantiation of $\mathbf{X}$ in $\theta_{\mathbf{X}}$ is irrelevant: taking a transition from s_i to s_j is independent of the observation emitted by s_i, since s_i is observed.

It is straightforward to verify that this construction produces a valid probability distribution, and it assures that $P'(\mathbf{X}^{(0:T)}) = P(\mathbf{X}^{(0:T)})$. As a side note, while $\mathrm{dom}(\mathbf{X})$ does not change in the simulated model $\mathcal{M}'$, this is not the case for $\mathrm{dom}(S)$, as opposed to the simulation of Proposition 3.1.

□

Proof of Proposition 3.3. The dynamic programming procedure for computing the expected statistics of one sequence stores the values of α in a $(T+1) \times k$ matrix, also known as lattice (or trellis) structure. The computation of a single α value has the cost of $\mathcal{O}(k+n)$ time as follows. In Equation 3.10, the summation amounts to $\mathcal{O}(k)$ as long as the transition distribution is encoded as a $k \times k$ matrix. The emissions in Equation 3.10, in turn, can be computed in $\mathcal{O}(n)$ time assuming

each state-specific BN is conveniently encoded (e.g. using a graph traversal with look-up tables for the parameters), allowing one to compute the probability of any joint event in linear time. Thus, the total cost for each α value is $\mathcal{O}(n+k)$, hence the entire lattice for one sequence takes $\mathcal{O}(Tk(k+n))$ time. We build a lattice for β as well, however the total cost per cell in this case changes to $\mathcal{O}(nk)$. Thus, the total cost for the lattice of β is $\mathcal{O}(Tk^2n)$.

Once the lattices for α and for β are done, we compute the expected statistic ξ. Based on Equation 3.9, computing one ξ value amounts to $\mathcal{O}(k^2n)$, thus an entire ξ lattice for one sequence takes $\mathcal{O}(Tk^3n)$ time. Finally, note that we can compute the expected statistic γ by means of $\gamma_t(i) = \sum_{j=1}^{k} \xi_t(i,j)$, thus the lattice for γ requires $\mathcal{O}(Tk^2)$ time by using the lattice of ξ.

The computation of all the lattices for an observation sequence is a sequential process in which the cost of ξ's lattice dominates over the rest. Hence, the total cost of one E-step iteration for one sequence in HMM-As amounts to $\mathcal{O}(Tk^3n)$. □

4

PREDICTING DISEASE DYNAMICS: A CASE STUDY OF PSYCHOTIC DEPRESSION

Unsupervised learning is often used to obtain insight into the underlying structure of medical data, but it is not always clear how to use such structure in an effective way. In this chapter, we apply structured hidden Markov models in order to build a probabilistic framework for predicting disease dynamics guided by latent states. The framework aims to facilitate the selection of hypotheses that might yield insight into the dynamics. We demonstrate this by using clinical trial data for psychotic depression treatment as a case study. The discovered latent structure and proposed outcome are then validated using standard depression criteria, and are shown to provide new insight into the heterogeneity of psychotic depression in terms of predictive symptoms for different interventions.

4.1 INTRODUCTION

Much about disease processes is unknown, as often the only available information about a disease are the patient's symptoms and signs. This might result in an incomplete understanding of a medical disorder, which can in many cases be overcome by latent variable modeling. In spite of requiring extra modeling efforts, latent variables can enhance our understanding of the problem domain by capturing unmeasured quantities (e.g. related to the underlying physiology) and their relationship to observed quantities [178], and might as well provide better fitted models [179]. Hence, by using latent variables, one can try to reconstruct the underlying structure of the process at hand by using observed data.

Unsupervised learning is the machine learning task that aims to generate representations of the underlying structure of the data. Well-established usage of unsupervised learning in medical data includes, e.g., the discovery of underlying patient groups using clustering methods [133, 134], which might help improve diagnosis and provide new insight into more effective treatment selection [2]. Other applications include feature selection from unlabeled data [105] where manual feature extraction might be not available or incomplete. Patient monitoring and alerting for the identification of clinical outliers has also been tackled by unsupervised techniques [83, 105]. Yet, when applied to medical data, unsupervised techniques generate output that often makes experts confront themselves with

questions like *what else can we do with this structure?*. This is particularly of interest in cases where it might be difficult to define hypotheses in advance to be tested, hence some form of exploratory data analysis must be conducted.

We show in this chapter that unsupervised learning methods, in particular hidden Markov models [141], can be used not only to describe the underlying structure but also to support the formulation of meaningful medical outcomes. Previous research suggested that the formulation of clinical outcomes might be guided by latent-variable models [96, 134], with the advantage of reducing the hypothesis space to be explored by inspecting model properties. By using HMMs, we claim that one can explore hypotheses on disease dynamics by inspecting model characteristics such as transition dynamics, latent states, etc.

In order to illustrate the usage of HMMs on disease dynamics, we make use of data from a clinical trial originally designed to compare pharmacological treatments to psychotic depression (PD) [176]. PD is a severe medical condition that is associated with a high burden of disease and relatively low remission rates following pharmacological treatment [147]. Although recent research has considered PD as a homogeneous subtype of major depressive disorder [177], the possibility that this subtype itself is heterogeneous should also be considered, which would stimulate the development of subgroup adjusted prognostics and treatment modifications. In this work, we apply HMMs to one of the largest pharmacological trials of patients with PD conducted so far [176], aiming to explore potential differences in course characteristics in the whole sample of patients and differences in sensitivity to treatment between medication groups.

The contributions of this chapter are as follows. We present a procedure to guide the exploration of hypotheses on disease dynamics by means of HMMs. We then apply this methodology to yield insight into the dynamics of PD treatments by exploring clinically meaningful outcomes. The hypotheses generated using the method are then tested based on standard clinical characterization of response and remission in PD. To the best of our knowledge, this is the first effort into a more systematic, data-driven approach for exploring hypotheses on disease dynamics based on probabilistic graphical models.

The remainder of this chapter is organized as follows. In Section 4.2 the relevant work related to this chapter is discussed. In Section 4.3 the proposed framework for exploring insight into latent disease dynamics is introduced. In Section 4.4 the psychotic depression data used as case study is described together with some descriptive statistics. In Section 4.5 the HMM proposed for modeling PD dynamics is detailed. The experimental results are shown in Section 4.6. The obtained results are validated in Section 4.7. Section 4.8 summarizes the chapter and gives suggestions for future work.

4.2 RELATED WORK

Probabilistic graphical models have been extensively used in medicine and psychiatry. Recently, network models have shown to provide new insight into depression and other disorders by exploring symptom pathways [16, 48]. These

models, however, do not employ latent variables and instead claim that disease complexity emerges from direct connections between symptoms. On the other hand, latent-variable models such as hidden Markov models have been also extensively used in medical domains. One advantage of HMMs is that one can easily incorporate domain knowledge into the model, e.g., by constraining model transitions and emissions [105].

When using HMMs to capture disease dynamics, it is often the case that the number of latent states is determined in advance, as researchers might be interested in a specific subset among all possible models. In [90] a two-state HMM has been used to investigate the hypothesis that patients switch between two stable states (symptom-free versus depressed) in major depressive disorder. To investigate the relationship between cognition and psychotic symptoms in Alzheimer's disease, in [154] a four-state continuous-time HMM as considered. By opposition, one might argue that by not imposing an *a priori* number of or already known latent states, a more ample set of possible models is considered, which can lead to more insight into disease dynamics, at the cost of a likely increased difficulty to interpret such models.

The typical usage of HMMs is in prediction or as a model to describe the underlying structure of the data [138]. While prediction is self-explanatory, the description of the underlying structure is often seen as a set of clusters, and for that reason it is a more abstract and more difficult representation to get insight from. A much more specialized usage of latent variables lies in the development of data-driven outcome measures, as suggested in [96, 134]. A data-driven approach to generating outcomes has the advantage that latent states might provide a more natural, compact and empirically-oriented way to measure multiple relationships between symptoms and other observables.

More recently, HMMs have been applied to electronic health records [93, 121], which concerns much larger (and often more heterogeneous) collections of data than usually seen before. Yet, such datasets are of very different nature and thus require new methodology for using models as HMMs for the discovery of relevant knowledge.

4.3 A PROBABILISTIC FRAMEWORK FOR CAPTURING DISEASE DYNAMICS

In this section we discuss models suitable for capturing latent disease dynamics in a probabilistic framework.

4.3.1 *Latent variable modeling*

In many problems, the measured variables reflect only part of the ongoing process as it is the case with disease symptoms, which can be seen as manifestations of some unobserved underlying disorder. Latent variables can be used to capture such unmeasured quantities and the way these relate to the observed ones [178], which results in a more complete model of the problem at hand, and might also

allow for a better model fit [179]. In temporal problems, such as clinical trials for patient treatment, one is also typically interested in the sequential relationship between latent states.

Hidden Markov models are models based on latent variables that are able to cope with uncertainty and sequential phenomena, which make HMMs suitable for many biomedical problems [90, 121, 154]. In this work we opt for modeling the observation space as a Bayesian network, which results in a model called *standard* HMM (see Chapter 3 for further detail). The standard HMM allows for more general representations of symptom interaction than the often used independent HMM, in which the symptoms are assumed conditionally independent given the latent state. By opposition, in standard HMMs the emission distribution is given by Equation 2.27. By modeling the observation space as a BN, more insight into the problem can be obtained by a more concise latent-state representation, as discussed in Chapter 3.

4.3.2 *State trajectories*

In modeling medical domains, we assume that there is a set of observable variables $\mathbf{X}$, where each variable $X_i \in \mathbf{X}$ will often refer to measured data such as symptoms, lab exams, medication, etc. We also model a latent variable S which represents states of the underlying disease (e.g. a disease remitting situation). The disease process of interest is assumed to be a discrete process over some time horizon, where the value of the latent variable and the observables that hold at time $t \geq 0$ will be denoted by $S^{(t)}$ and $X_i^{(t)}$ respectively.

HMMs can be used to predict the hidden states associated to observations, i.e. to compute the set of states that better explain the observations. The set of most likely states depends on the optimality criterion chosen according to the intended usage of such predictions [141]. In this chapter, we seek the states which are individually most likely, as we are interested in the chances that a patient will transition to one or more states that might represent, e.g., disease recovery. Hence, the average number of times a state is predicted to occur is the quantity of interest. This differs from the so-called Viterbi path, where one seeks the most likely state sequence jointly taken over some time horizon such as $\{0, \ldots, T\}$.

In order to predict the states which are individually most likely, one first computes the distribution of latent states at each time point t conditional on the complete patient's symptom data (i.e. the data over all the process duration). This is given by Equation 2.41 used in the Baum-Welch algorithm, which we repeat below for convenience:

$$\gamma_t(s) = P(S^{(t)} = s \mid X_1^{(0:T)}, \ldots, X_n^{(0:T)}) \tag{4.1}$$

After this has been done, the sequence of states for a given patient is obtained by selecting the most likely state at each time t:

$$\hat{s}_t = \arg\max_{s \in \mathrm{dom}(S)} \gamma_t(s) \tag{4.2}$$

for all $t \in \{0, \ldots, T\}$. This can be interpreted as *assigning* patients in states. For brevity sake, we do not index the predictions of Equation 4.2 by patient, although it should be clear that there is a set of predictions $\hat{s}_t$, $t \in \{0, \ldots, T\}$, for each patient.

4.3.3 *Exploring medical outcomes*

One way to obtain insight into disease dynamics is by considering transition dynamics between latent states. This is convenient because each latent state can take into account multiple symptom dimensions at once, which makes reasoning over patient trajectory very natural. Once the states are discovered, a detailed outcome measure that provides insight into treatment dynamics can be formulated.

We propose a procedure to build outcome measures in Figure 4.1. The procedure selects a set of *baseline states* $\mathcal{S}_b$ based on a selection criterion. From the remaining states, a set of *target states* $\mathcal{S}_e$ are to be selected based on its own criterion. Once $\mathcal{S}_b$ and $\mathcal{S}_e$ are obtained, *state reachabilities* from $\mathcal{S}_b$ states to $\mathcal{S}_e$ states are calculated. By varying the time interval between two given states of $\mathcal{S}_b$ and $\mathcal{S}_e$, the resulting probabilities $reach(i, j, t_1, t_2)$ indicate the temporal influence of a baseline state over a target state. Such state reachabilities can then be used to compose a rich outcome measure, e.g., by making $t_1 = 0$ and $t_2 \in \{1, \ldots, T\}$, which will result in a reachability trend as indicated in Figure 4.1.

4.3.4 *Selecting states*

The selection of baseline states of Figure 4.1 can be viewed in general terms as a function $f\colon \mathrm{dom}(S) \to \{0, 1\}$, as shown in Definition 4.1.

Definition 4.1 (Baseline state)**.** *We say that a latent state $s \in dom(S)$ is a baseline state if $f(s) = 1$. The set of baseline states is given by:*

$$\mathcal{S}_b = \{s \in dom(S)\colon f(s) = 1\} \tag{4.3}$$

The set of target states $\mathcal{S}_e$ of Figure 4.1 can be defined analogously.

We define in the following different criteria for selecting baseline and target states either by using model parameters or predicted patient trajectories (or both). These definitions can be seen as particular instantiations of the function f from Definition 4.1. We denote by D the set of patients, which typically corresponds to the data used to learn the model.

Definition 4.2 (Baseline-state criterion 1)**.** *We say that a latent state $s \in dom(S)$ is a baseline state if $\hat{s}_0 = s$ holds for at least one patient of D.*

In other words, Definition 4.2 labels a state as a baseline state if one or more patients are predicted to be in this state at the process start (i.e. at $t = 0$). A more general selection criterion of baseline states would specify a *degree of uncertainty* concerning the predictions made at baseline, as shown in Definition 4.3.

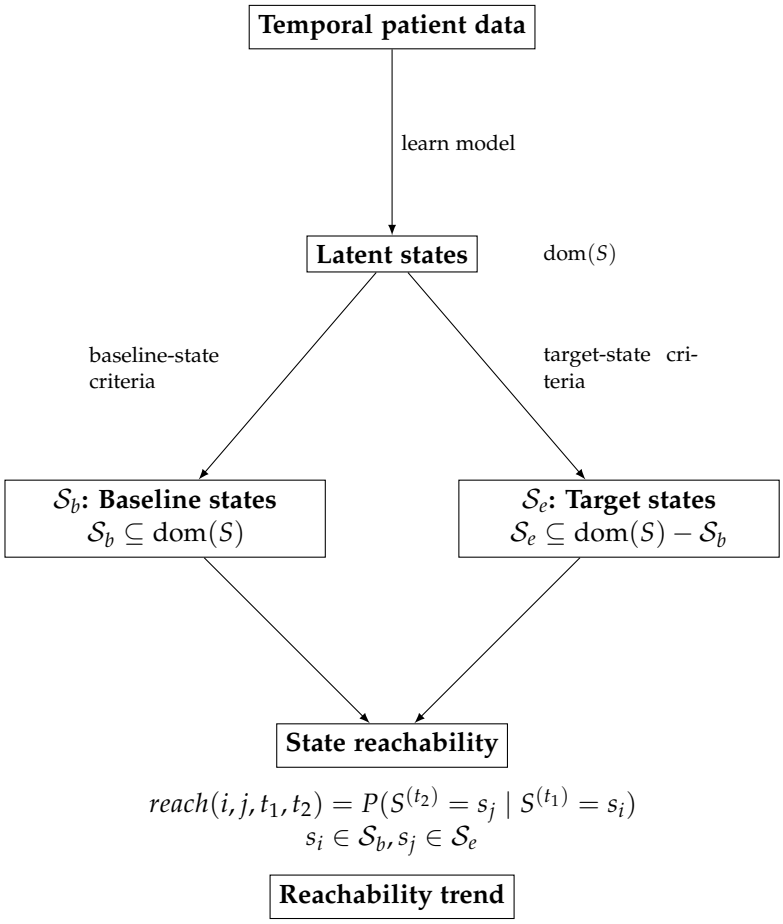


Figure 4.1: Procedure to guide the generation of outcome measures based on latent-state models.

Definition 4.3 (Baseline-state criterion 2)**.** *We say that a latent state $s \in dom(S)$ is a baseline state if all the following conditions hold for at least one patient of D:*

- $\hat{s}_0 = s$
- $\gamma_0(s) \geq \sigma$

where $0 < \sigma \leq 1$ is the minimal degree of uncertainty.

Definition 4.3 allows one to specify the minimal uncertainty on the state prediction that is acceptable. For example, with $\sigma = 0.95$, one imposes that the baseline state must have been predicted with low uncertainty at $t = 0$. This notion defines how strict one is for deeming a state as a baseline state. Note that

parameters such as the minimal degree of certainty and the minimum number of patients (the previous definitions required at least one patient) are part of the selection criterion and may be adjusted by the user.

For target states, Definition 4.4 presents a criterion based solely on model parameters.

Definition 4.4 (Target-state criterion)**.** *Let $s \in dom(S) - \mathcal{S}_b$ be a non-baseline state. We say that s is a target state if $P(s^{(t+1)} \mid s^{(t)}) \geq \rho$, where $0 < \rho \leq 1$.*

One can use Definition 4.4 by setting, e.g., $\rho = 0.95$, which would choose non-baseline states that have a high self-transition probability. Depending on the selection criteria, the target states could act as possible final states to the process at hand by representing different patient recovery in terms of symptom severity.

4.4 DATA

4.4.1 *Patients*

All patients had participated in the DUDG (Dutch University Depression Group) study [176], a 7 week double-blind randomized clinical trial originally designed for comparing the effectiveness of venlafaxine, imipramine and venlafaxine plus quetiapine (V+Q, for brevity) in PD. The dataset originally included 122 participants aged 18-65 who met DSM-IV-TR criteria for a unipolar major depressive episode with psychotic symptoms and a 17-item Hamilton Depression Rating Scale score (HAM-D, for short) [81] of at least 18, both at the screening visit and at baseline. Table 4.1 describes the symptom items used to compose the HAM-D score of each patient, which is obtained by summing the score on each item. The resultant HAMD-D score indicates severity of depression as follows: normal (0-7), mild depression (8-13), moderate depression (14-18), severe depression (19-22), and very severe depression (greater than or equal to 23).

Because of insufficient information about the specific nature of psychotic symptoms, three patients were not included in the current study resulting in a dataset with 119 patients. From the total group, 59 (49,6%) were females; the mean age was 51.1 (StDv 10.9) years. Forty patients were randomized to treatment with imipramine, 38 to venlafaxine and 41 to V+Q .

4.4.2 *Baseline and follow-up variables*

Severity of depression (HAM-D, represented as a continuous variable) and the presence of psychotic symptoms (each represented as a dichotomized variable) were measured at baseline (i.e. before treatment starts) and weekly thereafter. Psychotic symptoms are delusions and hallucinations (totals at baseline, 36 and 9 in imipramine, 37 and 11 in venlafaxine, and 38 and 9 in V+Q respectively). At baseline, mean [StDv] HAM-D scores were 32.5 [4.9] in imipramine, 31.7 [4.6] in venlafaxine, and 31.6 [5.4] in V+Q.

Item no.	Symptom	Score range
1	Depressed mood	0-4
2	Guilt	0-4
3	Suicide	0-4
4	Insomnia (initial)	0-2
5	Insomnia (middle)	0-2
6	Insomnia (delayed)	0-2
7	Work and interests	0-4
8	Retardation (psychomotor)	0-4
9	Agitation	0-2
10	Anxiety (psychic)	0-4
11	Anxiety (somatic)	0-4
12	Somatic symptoms (gastrointestinal)	0-2
13	Somatic symptoms (general)	0-2
14	Genital symptoms	0-2
15	Hypochondriasis	0-4
16	Loss of insight	0-2
17	Loss of weight	0-2

Table 4.1: Composition of the HAM-D score for depression. For any patient, the HAM-D score is obtained by the summing the scores of all the symptoms. The grading of items with range 0-4 is as follows: 0 - absent, 1 - mild or trivial, 2 and 3 - moderate, 4 - severe. For the other items, the grading is: 0 - absent, 1 - slight or doubtful, 2 - clearly present.

A total of 98 patients completed the trial (34 in imipramine, 30 in venlafaxine, and 34 in V+Q). Data on patients who dropped out was imputed following the last-observation-carried-forward approach, as in the original study [176].

4.4.3 *Depression assessment*

At the end of medical treatment, patients were assessed according to conventional criteria for response and remission of depression [176]. Response was defined as a reduction of at least 50% on the HAM-D score compared to baseline and a score of 14 or below, and remission as a score of 7 or below.

4.5 A MODEL FOR PSYCHOTIC DEPRESSION

In this section we introduce a model for capturing the temporal latent structure of psychotic depression treatment.

4.5.1 *General and intervention-specific model*

In order to unravel treatment dynamics of the full sample of patients, as well as specific intervention-based treatment dynamics, a set of hidden Markov models

are learned. The model learned from the full sample is referred to as the *general model*, while models learned for each intervention are called *specific models*.

In order to aid comparisons of model dynamics in terms of transitioning behavior, the specific and general models share the same latent states. To this end, the general model is estimated, then each specific model is set with the obtained latent states. Then, the transition probabilities of each model are estimated using the corresponding intervention-specific data.

4.5.2 *Model parameters and structure*

The observable variables in the HMM used in this work are modeled according to the BN shown in Figure 4.2, which allows for a more expressive representation than the naive-Bayes structure, because there are direct interaction between symptoms. By doing so, we impose less independence assumptions than the naive solution, thus the model becomes more flexible in that more dependences can be induced from data. For the PD problem, once in a state the observables are parameterized as follows: the psychotic symptoms are encoded as binary random variables, while the depressive symptom (the HAM-D score) is parameterized as a conditional Gaussian distribution, which is conditioned on the state and on both psychotic symptoms, as shown in Figure 4.2.

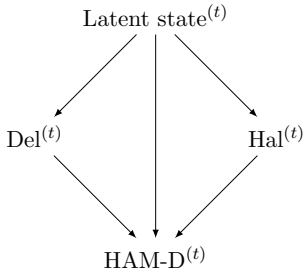


Figure 4.2: Graphical structure of the HMM emission distribution, showing the latent variable and its direct probabilistic influence on the observables at time t. Del and Hal denote delusions and hallucinations symptoms respectively, whose domains are {absent, present}, while the domain of the state variable is a positive integer which will be determined experimentally.

At any time point, the parameterization of each symptom is as shown in Tables 4.2 and 4.3. For a given state $s \in \text{dom}(S)$, the distribution of HAM-D can be obtained by marginalizing out Del and Hal and by applying the Bayesian network factorization as follows (we omit the time index as it is equal to t):

$$p(\text{HAM-D} \mid s) = \sum_{\text{Del,Hal}} p(\text{HAM-D}, \text{Hal}, \text{Del} \mid s) \tag{4.4}$$

$$= \sum_{\text{Del,Hal}} P(\text{Del}, \text{Hal} \mid s) p(\text{HAM-D} \mid \text{Del}, \text{Hal}, s) \tag{4.5}$$

As a result, the distribution of HAM-D conditional on state s is Gaussian as it is a linear combination of the Gaussians associated to the possible configurations of Del and Hal.

Variable	**Distribution**
Del	$P(\text{Del} = \textit{absent} \mid S = s)$
Hal	$P(\text{Hal} = \textit{absent} \mid S = s)$

Table 4.2: Parameterization of psychotic symptoms in the HMM emissions. Del, Hal and S denote delusions, hallucinations and state variables respectively. Note that $P(\text{Del} = \textit{present} \mid S = s) = 1 - P(\text{Del} = \textit{absent} \mid S = s)$ and similarly for Hal.

Distribution of HAM-D	**Parents** (plus some $S = s$)
HAM-D $\sim \mathcal{N}(\mu_1^s, \sigma_1^s)$	Del $= \textit{absent}$, Hal $= \textit{absent}$
HAM-D $\sim \mathcal{N}(\mu_2^s, \sigma_2^s)$	Del $= \textit{absent}$, Hal $= \textit{present}$
HAM-D $\sim \mathcal{N}(\mu_3^s, \sigma_3^s)$	Del $= \textit{present}$, Hal $= \textit{absent}$
HAM-D $\sim \mathcal{N}(\mu_4^s, \sigma_4^s)$	Del $= \textit{present}$, Hal $= \textit{present}$

Table 4.3: Parameterization of the HAM-D score in the HMM. The variable HAM-D is a mixture of Gaussian distributions of the form $\mathcal{N}(\mu_i^s, \sigma_i^s)$, where μ_i^s and σ_i^s denote the mean and standard deviation of the ith combination of parents given a latent state s, respectively.

Whenever the model is in a state, observations are emitted and a transition for the next time point is taken, and so on. The parameterization and structure discussed above are the same for all the specific models (i.e. the models obtained from each intervention data).

4.6 RESULTS

4.6.1 *Model dimension*

The number of latent states was obtained by balancing model fit and interpretability. Log-likelihoods were obtained from a 10-fold cross validation procedure, where models can have from two states up to the number of states obtained prior to model overfitting (see Appendix 4.A for more information).

The selected number of states considers the mean cross-validation fit and the corresponding confidence intervals shown in Figure 4.7, which is justified by the fact that in simpler models the role of latent states is more easily understood, because the states are likely more dissimilar in terms of associated symptom distribution and transition patterns. Also in favor of this procedure is the fact that the whole patient sample is split into treatment-specific data for model learning, which would make models with more states less stable. Appendix 4.A also shows scores of the Bayesian information criterion which support the selection based on cross validation.

4.6.2 *Identified states*

The general model has 3 latent states as shown in Figure 4.3 (top row), where in each latent state there is one distribution for each symptom measurement (i.e., Del, Hal and HAM-D). The states can be interpreted as follows:

- The **state Hallucinations (abbreviated as state *h*)** is associated with patients with high prevalence of hallucinations and moderate prevalence of delusions. Its mean HAM-D score is the highest among all states, while it has the narrowest tail.
- The **state Delusions (abbreviated as state *d*)** is associated with patients with high prevalence of delusions and low prevalence of hallucinations. Its mean HAM-D score is moderate and has wide tail.
- The **state No Psychosis (abbreviated as state *r*)** is associated with patients with low prevalence of psychotic symptoms and moderate HAM-D score (though with wide variance).

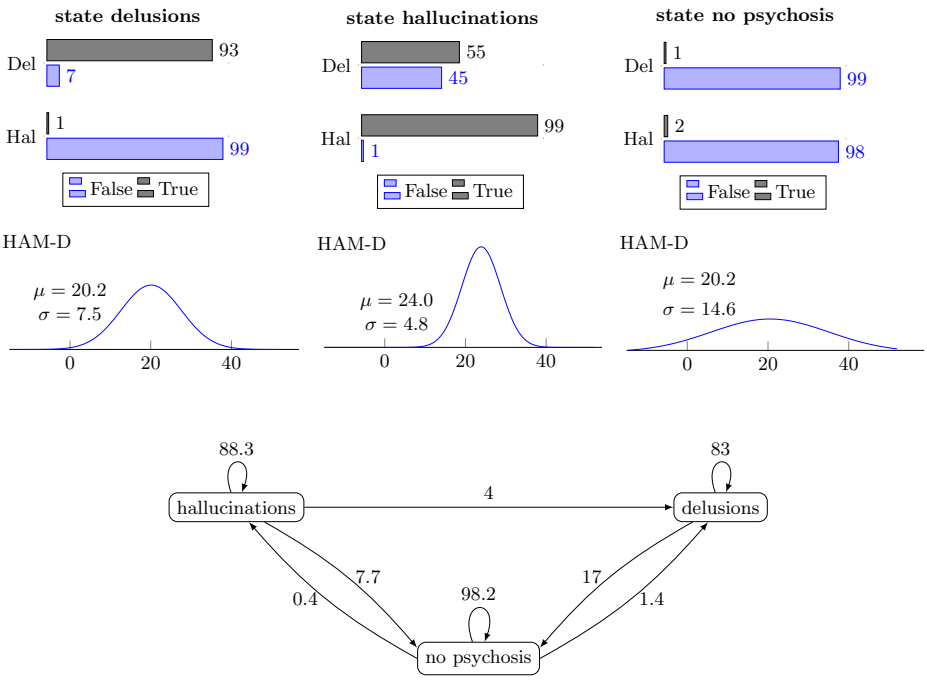


Figure 4.3: Top: marginal distributions of symptoms in the latent states of the general model (Del and Hal stand for delusions and hallucinations symptoms, respectively). Bottom: dynamics of the general model. Labels indicate transition probabilities between states (in %).

4.6.3 *Dynamics*

Figure 4.3 (bottom row) shows the transition behavior of the general model. The arcs indicate transition probabilities between latent states, e.g., the looping probability of 88.3% in state h represents the chance for reiterating in such state over two adjacent weeks. Based on Figure 4.3 (top row) and on the previous characterization of the states, d and h can be seen as starting states that are primarily distinguished based on the prevalence of hallucinations in a patient. Later on, depending on the patient's response to treatment, the patient will potentially move to state r. The state r can be seen as a healthier state due to the absence of psychotic symptoms, but such state does not imply depression remission or response due to its moderate mean HAM-D. In fact, the state r characterizes a wide range of no-psychosis patients, from those that still have high HAM-D to those that have achieved low HAM-D.

4.6.4 *Comparing interventions*

From the obtained latent states shown in Figure 4.3, we now detail an outcome measure based on the procedure established in Section 4.3.3, which will also allow for comparing interventions. Based on state trajectories (Equations 4.1 and 4.2), at baseline 90 patients were assigned to state d with mean (StDv) probability of 100% (0), while 29 patients were assigned to state h with mean (StDv) probability of 93.6% (13.2%). Hence, very little uncertainty was entailed by the model as to which initial state any given patient is predicted to be in. As a consequence, the criteria specified in Definitions 4.2 and 4.3 coincide for the PD study case, resulting in the set of baseline states $\mathcal{S}_b = \{d, h\}$. As for the set of target states $\mathcal{S}_e$, Figure 4.3 shows that the state r has a self-transition probability of 98.2%, which allows us to set $\mathcal{S}_e = \{r\}$.

Given the sets of states $\mathcal{S}_b$ and $\mathcal{S}_e$, we define the reachability as the chances to reach the state r at time t_2 from one of the baseline states at $t_1 = 0$:

$$reach(i, j, t_1, t_2) = P(S^{(t_2)} = s_j \mid S^{(t_1)} = s_i) \tag{4.6}$$

$$reach(b, r, 0, t_2) = P(S^{(t_2)} = r \mid S^{(0)} = b) \tag{4.7}$$

where b is either the state d or the state h, and $t_2 \in \{1, \ldots, 7\}$.

In order to compare interventions, reachability values were computed from the general model (Figure 4.3), as well as from the specific models (see Appendix 4.B for more detail on the specific models). The obtained reachability values were made further robust by a bagging procedure [20], where models are learned from bootstrap samples to provide more stable outcome measures. In this work, 10,000 bootstrap samples were generated, a model learned from each one and the corresponding reachability values computed. The reachability trend provided by the models learned from the bootstrap samples are then used to compute confidence intervals that indicate the variability of the reachability trend. In the following this idea will be further explored for comparing the general and

specific models learned from the full sample and from each intervention data, respectively.

4.6.5 *Reachability trend per treatment*

Figure 4.4 shows the reachability trends grouped by intervention. The difference between the area under the curve (AUC, for short) of each trend was also computed. For the whole sample of patients, the 95% (BCI, for short) of the AUC difference was [0.17, 2.29], while for the slope difference the AUC was [0.02, 0.17], where positive values indicate a stronger trend in favor of state d. Under venlafaxine, the AUC difference was [0.16, 3.09], whereas the slope difference was [0.01, 0.23]. These results suggest that the initial state of the patient is relevant under venlafaxine in that starting in state d allowed for a significantly stronger reachability towards state r than the reachability had the patient started in state h.

Under imipramine, the AUC difference was [-1.62, 2.36] and the slope difference was [-0.15, 0.19]. Finally, for V+Q the AUC difference was [-0.74, 3.72], and [-0.10, 0.32] for the slope difference. Hence, starting in state d for imipramine and for V+Q also provided stronger trends towards r, but not to a significant extent. The detailed difference BCIs per week can be found in Appendix 4.C.

4.6.6 *Reachability trend per starting state*

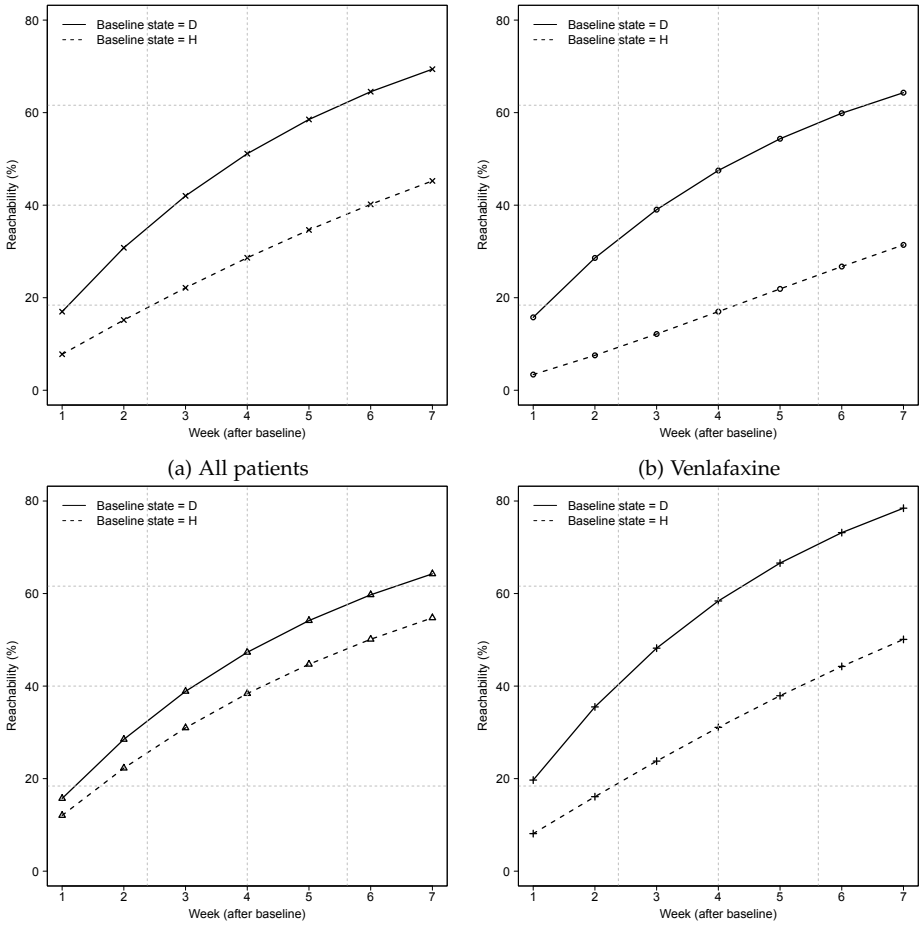
Figure 4.5 shows the reachability trends of Figure 4.4, now grouped by starting state. Patients can either start in state d (Figure 4.5-a) or in state h (Figure 4.5-b). Figure 4.5-a suggests that if a patient had no hallucinations at baseline (i.e. started in state d), then a stronger reachability trend would be achieved if this patient were treated with V+Q. For patients that had experienced hallucinations (i.e. started in state h), the results suggest that the strongest trend would be achieved with imipramine. Nevertheless, 95% BCIs indicate that no significant differences were found when comparing the trends starting in h, nor when comparing those starting in d.

4.7 VALIDATION

In this section we investigate if aspects of the learned model and the formulated outcome can be associated to standard depression criteria computed directly from the data, as means to validate the model and the outcome.

4.7.1 *Model validation*

Associations between model outputs in the form of state trajectories (see Section 4.3.2) and depression recovery (see Section 4.4.3) were computed. For each patient, we counted the number of consecutive weeks in which state r was predicted



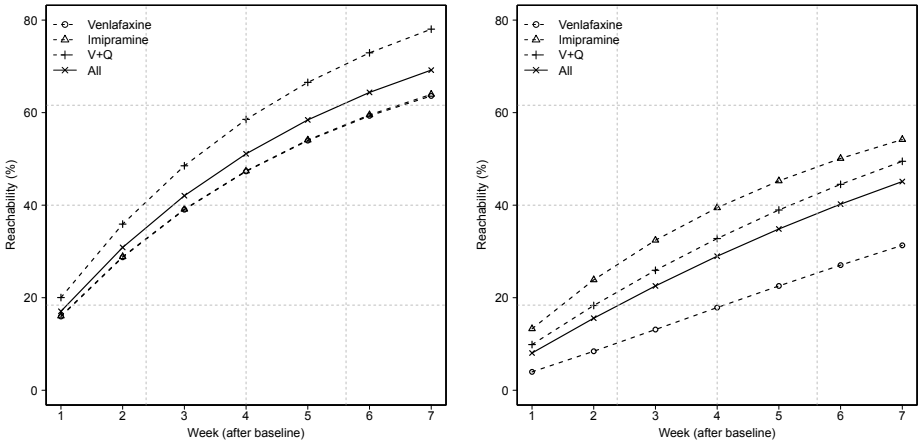
(a) All patients (b) Venlafaxine

(c) Imipramine (d) V+Q

Figure 4.4: Reachability trends per intervention.

as the most likely state (see Equation 4.2). In case the endpoint of patient state trajectory is not predicted as state r, the assigned count is zero. Among the total sample, 60 patients achieved depression response, with the state r predicted in 4.7 weeks on the average, while the 59 patients who did not achieve response had the state r predicted in 1.3 weeks on the average. Figure 4.6 shows a histogram of the number of patients versus the number of consecutive weeks for which state r was predicted. A Fisher's exact test was applied to compare the counts of the two groups from Figure 4.6 (responders versus non-responders), which resulted in a p-value < 0.001, suggesting that these two groups (responders and non-responders) associate significantly different to the number of weeks in the state r (under a 95% confidence level).

Among the total sample, 35 patients achieved depression remission, with the state r predicted in 5.4 weeks on the average, while the 84 who did not achieve



(a) At baseline: state d (b) At baseline: state h

Figure 4.5: Reachability trends per latent state. The Y axis denotes the reachability at each week after baseline.

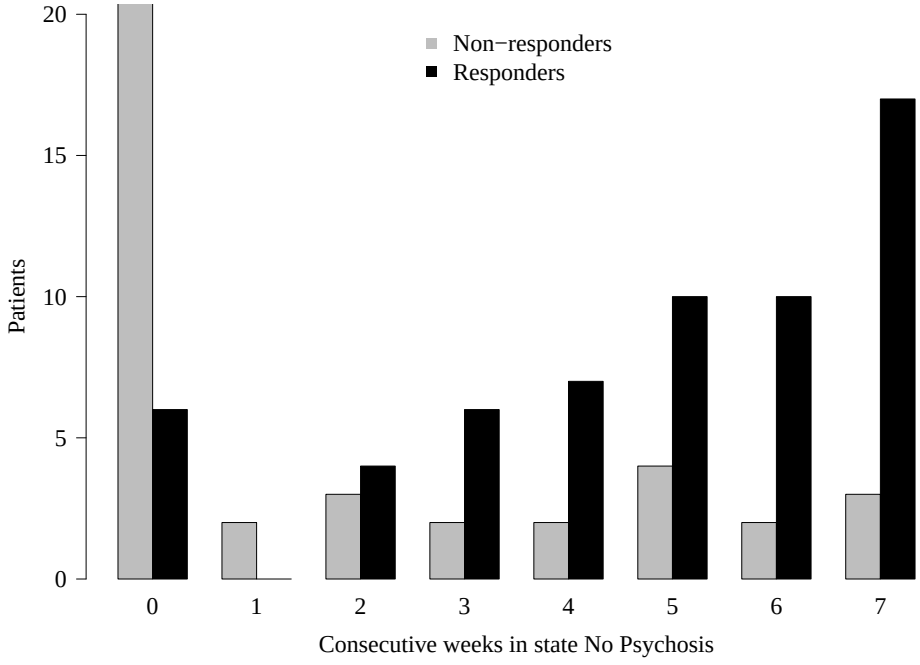


Figure 4.6: Histogram of the number of times the state r was predicted in patient state trajectory. The two groups refer to patients who achieved depression response (60 patients) and those who did not (59 patients). For the sake of visualization, zero consecutive weeks for non-responders was cut down (original value was 41 patients).

remission had the state r predicted in 2.0 weeks on the average. A Fisher's exact test to compare remitters versus non-remitters resulted in a p-value < 0.001 (histograms for remitters were omitted due to the small numbers). These results support the claim that the state r is meaningful in terms of distinguishing patients that achieved depression recovery (either response or remission) from those who did not.

4.7.2 *Outcome validation*

We now assess the claim of Section 4.6.5 that the state at baseline leads to significantly different state reachability for the total sample case. To this end, two distinct groups of patients were considered: patients with hallucinations at baseline (29 patients, see Section 4.4.2), and patients with no hallucinations at baseline (90 patients). The HAM-D scores of these groups at treatment endpoint were compared using a Mann-Whitney test for independent samples, which resulted in a p-value = 0.0007, thus suggesting that these two groups differ significantly (under a 95% confidence level). As a consequence, the psychotic symptom at baseline is predictive to depression recovery of patients in general. This evidence supports the conclusions for the model-based outcome drawn in Section 4.6.5, where the psychotic symptom at baseline was found to be predictive to reaching the state r when one considers all the patients (Figure 4.4-a).

4.8 CONCLUSIONS

This chapter demonstrated that probabilistic graphical models can reveal insight into disease dynamics by considering not only the underlying structure, but also by looking at meaningful outcome measures built from such structure. We illustrated the proposed methodology by applying hidden Markov models to psychotic depression treatment data, where the models were learned in a fully data-driven way.

The identified temporal symptom structure of psychotic depression revealed that patients differed in their prognosis depending on the type of psychotic symptoms they exhibited at baseline (hallucinations versus delusions). This result was observed for the total sample and for the patients that underwent venlafaxine intervention. Hence, our methodology allowed to shed light on the heterogeneity of psychotic depression. As future work, we plan to further investigate the clinical significance of the results, as well as consider the effect of potential confounders, such as patient demographic data.

The combination of graphical models and a data-driven approach can be easily integrated into the investigation of other psychiatric disorders as well, potentially helping physicians to understand disease dynamics and may even support them in prescribing optimal pharmacological therapy. Furthermore, by applying the proposed methodology to other diseases, it should be possible to assess the method more broadly. It could be of interest to perform different calculations of

state trajectories that reflects the availability of only partial symptom data (e.g. to simulate an ongoing treatment), or even calculate state reachability from different starting points other than the baseline point. One could also consider adding intermediate states to the proposed framework, which could allow for greater flexibility in situations where many more latent states are obtained.

4.A MODEL SELECTION SCORES

Figure 4.7 shows 10-fold cross-validation mean log-likelihoods for different number of latent states, together with 95% confidence intervals. The higher the log-likelihood of a model the better fitted such model is.

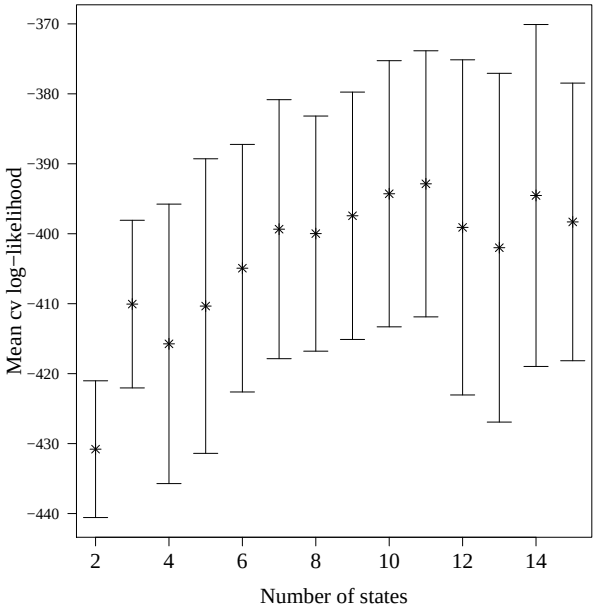


Figure 4.7: 95% CIs for the mean cross-validation log-likelihoods for selecting the number of states of the general HMM.

A Mann-Whitney test was performed for comparing the cross-validation log-likelihoods shown in Figure 4.7 of the 3 state model with that of other models. The resultant p-values (number of states) were: 1.0 (4), 0.91 (5), 0.48 (6), 0.31 (7), 0.35 (8); the maximum p-value of the remainder cases (9 up to 15 states) was 0.08.

In addition to the 10-fold cross validation results, BIC scores (Equation 2.14) were computed for different number of states, which balances goodness of fit with a penalty based on the number of parameters of the model and the sample size. Figure 4.8 shows the BIC scores for different models, suggesting the 3-state model achieves the minimal model selection score. This is in line with Figure 4.7, where the overlapping confidence intervals suggest that it is likely not significant

the improvement achieved by models with more than 3 states, hence a suitable dimension would be 3 states.

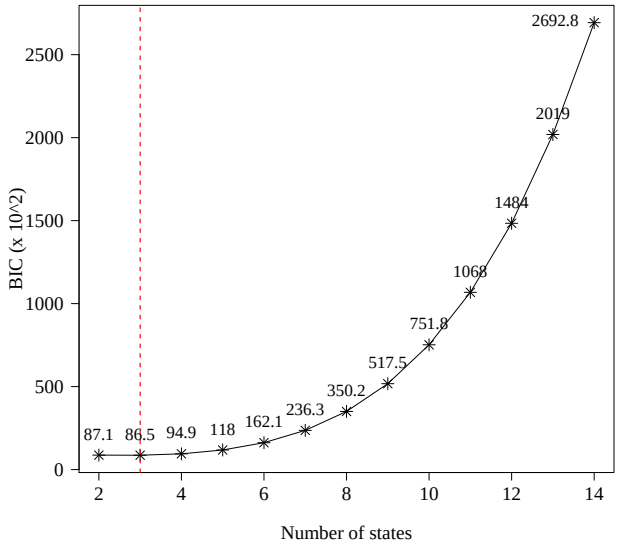


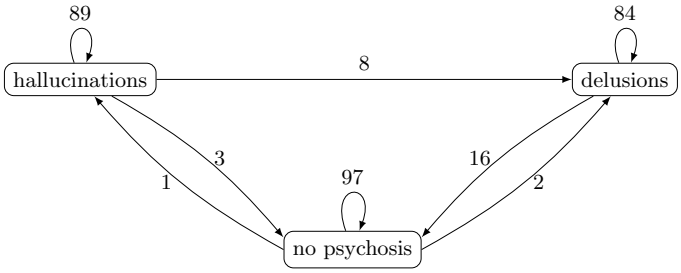
Figure 4.8: BIC scores of models with different number of latent states. The vertical dashed line indicates the number of states which led to the minimal BIC. We seek models that minimize the BIC.

4.B DYNAMICS OF INTERVENTION-SPECIFIC MODELS

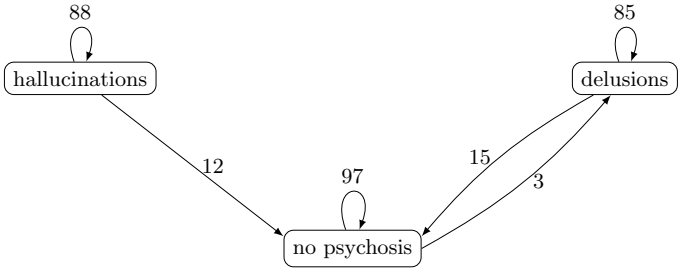
Figure 4.9 shows the transition probabilities of each intervention-specific model. As described in Section 4.5.1, all the specific models and the general model share the same latent states, which are shown in Figure 4.3 (top row).

4.C CONFIDENCE INTERVALS OF REACHABILITY TREND DIFFERENCES

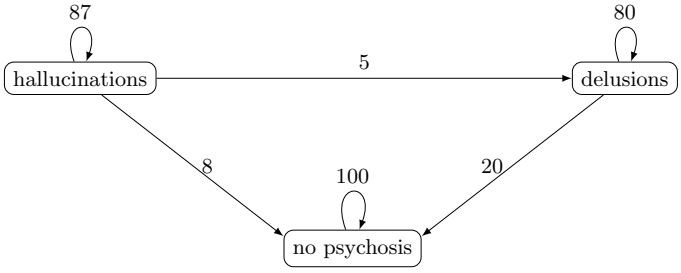
Figure 4.10 shows 95% bootstrap confidence intervals for the differences between the reachability trends of Figure 4.4.



(a) Venlafaxine

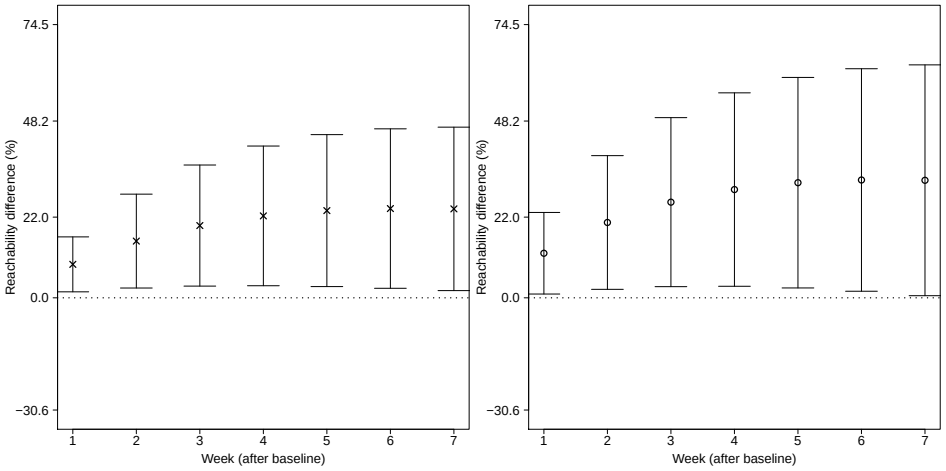


(b) Imipramine



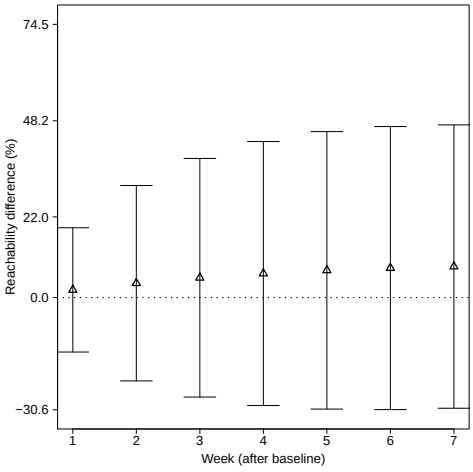
(c) V+Q

Figure 4.9: Dynamics of the intervention-specific models. Labels indicate transition probabilities between states.

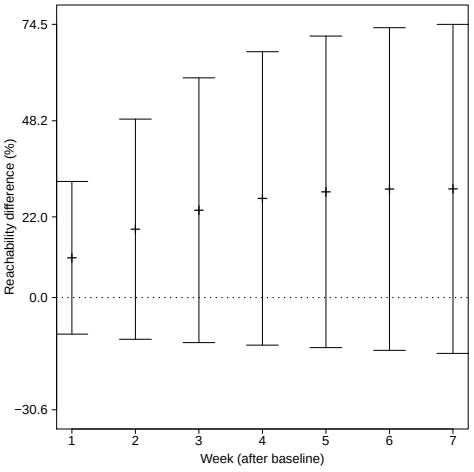


(a) All patients

(b) Venlafaxine



(c) Imipramine



(d) V+Q

Figure 4.10: 95% bootstrap CIs for the differences between reachability trends. The dotted line indicates a difference equal to zero. Positive values indicate higher reachability of state d compared to that of state h.

5

UNDERSTANDING MULTIMORBIDITY THROUGH CLUSTERS OF HIDDEN STATES

Nowadays, a significant portion of the population has more than one chronic disease at the same time, which is known as the problem of multimorbidity. *Better understanding multimorbidity is hindered by the fact that most available clinical research datasets are small in size, making it harder to investigate interactions between diseases. The current availability of large volumes of routinely collected health care data is a promising source for learning about disease interaction. In this chapter, we propose a latent or hidden variable-based approach to understand patient evolution in temporal electronic health records, which can be uninformative due to the fact that it contains little detailed information. We introduce the notion of* clusters of hidden states *which may allow for an expanded understanding of the multiple dynamics that underlie events in such data. Clusters are defined as part of hidden Markov models learned from such data, where the number of hidden states is not known beforehand. We evaluate the proposed approach based on a large dataset from Dutch practices of patients that had events on medical conditions related to atherosclerosis. The discovered clusters are further correlated to medical outcomes in order to show the usefulness of the proposed method.*

5.1 INTRODUCTION

With the availability of large volumes of health care data, promising new data sources have come to the disposal of the research community to investigate health care problems that require much data. A typical example is the study of interactions among diseases as done in *multimorbidity* research, i.e. when multiple diseases occur at the same time in people [6, 140, 159]. Influenced by factors such as the aging of the population, multimorbidity is the rule, not the exception. Mutimorbidity research is not really feasible with typical clinical research datasets, which are small in size and usually only deal with a single disease. More recently, machine learning techniques applied to electronic health records (EHRs, for short) in the order of billion data points have been able to provide accurate predictions [142], which shows that it is possible to take advantage of such datasets, despite their low quality compared to research datasets such as those from clinical trials.

In spite of its volume-related advantages, health care data are noisy, incomplete, and usually not directly suitable for research purposes, making analysis hard. One source of data used for investigating multimorbidity and disease interaction is data collected from visits to general practitioners [106], where each patient visit is often assigned a single diagnosis code meant for administrative and billing purposes. It is, however, possible that patients have additional conditions at the time of the visit (some of which might be chronic conditions, such as hypertension or Alzheimer's disease), which would mean the existence of multimorbidity in patient. It is also often the case that symptoms and signs are not available in such health care data. As a result, one cannot directly detect multimorbidity by simply looking at GP visits individually.

With health care data, one can resort to investigating sequential disease interaction in order to partially overcome the discussed limitations of such data. By doing so, one could ultimately obtain insight on multimorbidity. Uncertainty also plays a central role because future events are typically not completely determined by the current patient status. Much research has been dedicated to the analysis of health care data, but most of it tends to focus on managerial aspects such as patient flow, hospital resources, etc. [45, 120] more often than on understanding diseases dynamics [92, 126].

In this chapter, we hypothesize that using latent information next to the diagnostic data can increase our understanding of disease interaction dynamics. By using as a basis hidden Markov models [141], multiple latent states can be associated to a given diagnostic event (where an event could be a visit due to, e.g., type 2 diabetes mellitus or a myocardial infarction). Based on this, we introduce the notion of *clusters of hidden states*, where a cluster contains all the states that produce the same observation (i.e. the same event). Although apparently simplistic, states within a cluster can have quite different dynamics in terms of transitioning patterns (i.e. how a state can be reached by or left from). By looking at these transition patterns, we will be able to give multiple roles to each event, which sheds light on the influence of such event on disease interaction. Besides the structural differences of states within a cluster, we show that these states are associated in different ways to medical outcomes. The identification of latent information has been shown valuable for gaining a better understanding of health care data [91, 92], although we pursue a different angle on what to cluster than previous research.

The contributions of this chapter are as follows. We first define the notion of clusters of states from the perspective of electronic health records. This is followed by the identification of general transition patterns that might emerge in clusters of hidden states. We then introduce a case study based on data collected from Dutch practices amounting to 32,227 patients that had visits related to atherosclerosis. Atherosclerosis is a medical condition that can be seen as an umbrella term of many other diseases, thus it is suitable for illustrating clusters and the role of their states in real-world data. Once an HMM is learned from the atherosclerosis data, we provide application-oriented interpretation to the

clusters of states by looking at a medical outcome (the number of total diseases that were registered in patients) correlated to states of clusters.

This chapter is organized as follows. Section 5.2 describes the structure of EHRs and modeling assumptions. Section 5.3 defines clusters of states and transition patterns associated to them. Section 5.4 describes the data used as case study, while in Section 5.5 the results of applying the proposed notions of state clusters to such data are discussed. Section 5.6 discusses the related work, while Section 5.7 summarizes the chapter and discusses future work.

5.2 HEALTH-CARE EVENT DATA

5.2.1 *Representation*

Let us suppose that there are n possible diagnoses, each one represented by a random variable X_i taking values from the domain $\{0,1\}$, with $X_i = 1$ indicating *presence* and $X_i = 0$ *absence* of diagnosis i. The full set of diagnosis variables is denoted by $\mathbf{X} = \{X_1, \ldots, X_n\}$. This representation allows one to represent the occurrence of multiple conditions in patients at each time point. In the considered EHRs, however, patient visits to their general practitioner are recorded such that each patient visit is typically assigned a single diagnosis code (sometimes called the *main diagnosis*), which means that effectively only one disease is registered at each time point. The main diagnosis code in patient visits can be related, e.g., to a chronic condition (e.g. diabetes mellitus) or not (e.g. a fracture).

By taking the single diagnosis assumption into account, each event can be represented by an instantiation of $\mathbf{X}$, such that $X_i = 1$ and $X_1 = \cdots = X_{i-i} = X_{i+1} = \ldots = X_n = 0$, where X_i corresponds to the main diagnosis associated to the event. The time interval between any two visits is often arbitrary. Next to the diagnosis data, additional data might be available, such as medication prescription and results of lab exams.

An alternative representation would use a single variable taking values on a domain with n values, which could be seen as the state space of a Markov chain. However, we prefer using individual diagnosis variables because it is more general and flexible enough for easily allowing one to add more patient information into event data if such information is available. For example, if it is known that a chronic condition previously diagnosed still occurs in the patient, one could mark the corresponding variable as active in addition to the main diagnosis of the current visit. However, additional assumptions or patient data would be required in order to confirm such previous diagnoses, as there is always some degree of uncertainty as to whether previous conditions are indeed chronic. As a consequence, we did not make such assumptions.

5.2.2 *Modeling*

Health care data from EHRs is often fine grained, in the sense that each event will likely reflect only information that is limited to the current patient visit. This differs, e.g., from longitudinal clinical trials [175], which are often characterized by repeated measurements of symptoms and signs associated to one or more conditions. As a consequence, data from such clinical trials normally allows for a more complete assessment of patient evolution, as opposed to health care data. This suggests that one could capture unmeasured patient information in such EHR data by including *latent variables*, such that it could provide a richer characterization of patients when combined with observable data.

In this work, hidden Markov models are used to capture the sequential interaction between observable and latent variables. In the multimorbidity context, the diagnosis variables $\mathbf{X}$ correspond to the observable variables, and we assume that there is a latent variable S. The usage of hidden states attempts to compensate for the mentioned difficulties present in temporal EHRs. We consider the family of independent HMMs for modeling (see Chapter 2 for details on HMMs). This choice is justified by the large amount of data in EHR datasets and the low number of observable variables (as shall be discussed in Section 5.4).

In order to comply with the event data representation, we further assume that the emission distributions of the HMM are deterministic such that only one observable variable X_i is *active*, i.e. for every S there is some X_i such that:

$$P\left(X_j^{(t)} = 1 \mid S^{(t)}\right) = \begin{cases} 1 & \text{if } j = i \\ 0 & \text{otherwise} \end{cases} \tag{5.1}$$

5.3 IDENTIFYING TRANSITION PATTERNS

5.3.1 *Clusters of states*

The events constructed from health care data imply that in order to fully comply with the data concerning n diagnoses, the hidden states should be constrained to emit one out of n different observations at each moment, as defined in Equation 5.1. In spite of this apparent simplicity, the underlying process being modeled could still be quite complex (e.g. by having multiple stages at different moments). In order to properly capture such distribution, more states could be needed, which can lead to the situation where multiple states are associated to the same diagnosis (e.g. if one decides to model more states than observable variables). From these considerations, we define a *cluster of states* as a set of states that have the same emission distribution.

5.3.2 *Transition patterns*

Modeling state transitions in a probabilistic way, e.g. as in Markov chains, implies that a state can often be reached in different ways and can lead to different future

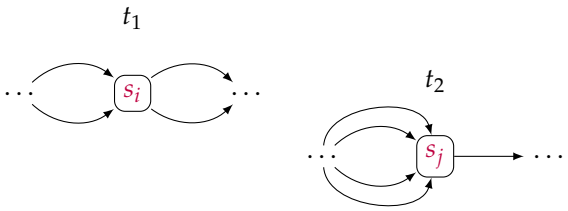


Figure 5.1: Cluster of states $C = \{s_i, s_j\}$, where s_i can be reached from two states and can transition to two states, while s_j can be reached from four states and can transition to a single state.

states. As we show next, by considering clusters of states such dynamics are further enriched, because such past-present-future transitioning can occur in multiple ways. For example, consider two states s_i and s_j belonging to a cluster C, as shown in Figure 5.1. This suggests that s_i will likely be reached earlier for the first time than s_j, and it also suggests that both states can lead to quite different incoming and outgoing states. Of course, such multiple *roles* of a given diagnosis (represented by the cluster C) stem from the complexity of the underlying process, where a given diagnosis could be associated to different medical situations when one looks at the whole care process. For example, the states of a cluster could be associated to different levels of severity or worsening of patient health that could happen at different moments.

In order to better understand the roles of states in clusters, we discuss transition patterns that might arise. This characterization involves states and transitions from and to them, and is provided at a high level, because it is intuitively unfeasible to anticipate all the possible ways by which the states of clusters can interact.

5.3.2.1 *Internal patterns*

A state is associated to an *internal transition pattern* if most of the probability mass of its incoming and outgoing probabilities associates to states from the same cluster. The most trivial internal pattern occurs when a state has a loop probability close to 1, which we call a *recurrent pattern*. A more formal description is that a state s has a recurrent pattern if s has a transition probability $P(S^{(t+1)} = s \mid S^{(t)} = s) \geq \alpha$, where α will typically be close to 1.

A more complex internal pattern would occur when there is a cycle involving two or more states from the same cluster. In this case, at any moment it is very likely that the system (e.g. a patient) is switching between the same diagnosis represented by different states. We call such patterns *internal feedback patterns*.

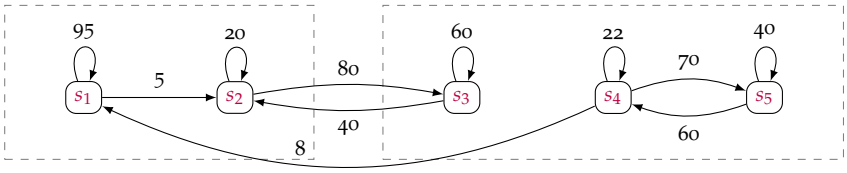


Figure 5.2: An example with two clusters of states $C1$ (left) and $C2$ (right) for depicting patterns of state transition. Probabilities are given by percentages.

5.3.2.2 *External patterns*

External transition patterns involve states from two or more clusters. One type of such patterns are the *external feedback patterns*, which involve states from two or more clusters such that most of the incoming and outgoing probabilities stay in the cluster.

In the context of disease interaction, external patterns occur when transitions involve different diagnoses, as opposed to internal patterns. Hence, if a cluster is involved in both an internal and an external pattern, then the same diagnosis could lead to different future events. In other words, the same diagnosis could play distinct roles.

Example 5.1. *Suppose two clusters of states* $C1 = \{s_1, s_2\}$ *and* $C2 = \{s_3, s_4, s_5\}$*, where* $C1$ *and* $C2$ *are associated to two different diagnosis codes, as shown in Figure 5.2. It holds that state* s_1 *is involved in a recurrent pattern due to its high self-transition probability (for* $\alpha = 0.95$*). States* s_4 *and* s_5 *are involved in an internal feedback pattern, while states* s_2 *and* s_3 *are involved in an external pattern.*

5.4 CASE STUDY

In order to illustrate the value of the proposed methods, we consider the Primary Care Database from the NIVEL institute (Netherlands Institute for Health Services Research), a Dutch institute that maintains routinely electronic health records from health care providers to monitor health in Dutch patients [127]. In the NIVEL data, patient visits are assigned an ICPC code (International Classification of Primary Care) indicating a diagnosis for the visit.

5.4.1 *Variables and observations*

We focus on variables related to atherosclerosis, which is a cardiovascular condition that has complex associations to a number of other conditions. Although in the literature atherosclerosis has been known to be associated to chronic diseases like diabetes [95], there is still active research on its implications and associations [125, 129, 164]. In our data pre-processing steps, we first selected ICPC codes related to atherosclerosis, then groups of codes that refer to a given medical symptom or condition were built based on medical experts. As a result, each

ICPC code, description	Variable (model)
K02.00, Pressure/tightness of heart K74.00, Angina pectoris K74.02, Stable angina pectoris K76.01, Coronary sclerosis	*Angina*
K75.00, Acute myocardial infarction K76.02, Previous myocardial infarction (> 4 weeks earlier)	*Myocardial infarction*
K89.00, Transient cerebral ischemia/TIA K90.00, Cerebrovascular accident K90.03, Cerebral infarct	*Cerebrovascular accident*
K92.01, Intermittent claudication	*Claudication*
K99.01, Aortic aneurysm	*Aortic aneurysm*
K91.00, Atherosclerosis	*Atherosclerosis*

Table 5.1: ICPC codes related to atherosclerosis, and their mapping into variables of the model.

group of codes gave rise to an observable variable, as shown in Table 5.1. The variables constructed based on Table 5.1 can be seen as comorbidities that might occur in patients with atherosclerosis.

In order to construct the event data from the raw NIVEL data, we first ordered the raw data in ascending dates. Then, whenever a patient visit having as diagnosis one of the ICPC codes from Table 5.1 was found, a new observation was created, where the variable associated to the ICPC code was instantiated as the value 1 and the remaining variables were assigned zeros. The visits that were not associated to any of such ICPC codes were ignored.

5.4.2 *Sample*

We considered a sample of 32,227 patients that had visits between 1st of January, 2003 and 31st of December, 2011. To be included, a patient must have had at least one visit related to one of the diagnoses listed in Table 5.1. The data construction procedure previously discussed resulted in a dataset with 216,580 observations, where the average number of observations per patient is 6.7 (StDv = 10.9). A total of 11,932 patients have only one observation, whereas 20,295 patients have two or more.

5.4.3 *Number of hidden states*

In order to select an appropriate number of states when learning HMMs, the Akaike Information Criterion (AIC, for short)

$$\mathrm{AIC}(M) = 2\log K - 2\log \hat{\mathcal{L}}(M) \tag{5.2}$$

was used, where M is a candidate model, K is the number of parameters of M, and $\hat{\mathcal{L}}(M)$ is the likelihood of M based on maximum likelihood estimates of the parameters.

The AIC is a less conservative model selection functions than scoring functions as BIC (see Section 2.14). This is justified in this situation because there are large amounts of data in the case study, which allows us to model more latent states by using the AIC score. The AIC score is supposed to be minimized. Models are evaluated by increasing their number of states until the addition of states does not improve the score substantially, which is an strategy to combat overfitting.

For learning of HMMs the Baum-Welch algorithm is used (see Section 2.6.2), which is sensitive to its initial parameters, especially with larger number of states. In order to reduce such effect, the best initial model was selected out of 30 candidates randomly generated.

5.4.4 *Clinical interpretation of clusters*

If clusters of states are identified in the learned model, one would expect that states within a cluster are indeed necessary, i.e. they should not be replaced by a single state, at the cost of, e.g, worsening model fit. The clusters of states and associated transition patterns also give insight in the *structural* role played by the states. In order to further understand the role of states of a cluster, we consider measures used in multimorbidity research. Multimorbidity measures can be used to look at patients from different angles, which is related to the notion of complexity of patient [117].

The most common way to measure multimorbidity impact in a population is by means of *disease counts* [94], in which single diseases are added resulting in a total number of diseases per patient. The count of diseases is related to the functional status and quality of life [94], thus it can be used to provide additional significance to the HMM states learned from the EHRs data. In this case study, the disease counts were calculated as the total number of distinct diagnoses that were registered for each patient, which might include other events than those listed in Table 5.1. This provides an approximation to the number of diseases that have occurred in the patient. We detail next the manner by which disease counts are associated to the latent states.

Let us consider a latent state $s_j \in \mathrm{dom}(S)$ and the ith patient in the data. We first compute the chances that this patient is in state s_j at some instant t based on the full observations of the patient, which is denoted by:

$$\gamma_t[i](j) = P(S^{(t)} = s_j \mid \mathbf{X}[i]^{(0:T_i)}) \tag{5.3}$$

where T_i refers to the last observation of the ith patient (see Section 2.6.2 for HMM notation). When the patient has more than one observation, this will result

in a sequence of probabilities for a state s_j. As we will associate the states to the total number of diseases, the average of such probabilities is taken:

$$\overline{\gamma}[i](j) = \frac{1}{T_i + 1} \sum_{t=0}^{T_i} \gamma_t[i](j) \tag{5.4}$$

From Equation 5.3, if the latent variable has k states $\{s_1, \ldots, s_k\}$, then each patient will be associated to k average state probabilities, one for each state s_j. It is straightforward to see that these average probabilities sum to 1.

Once the quantities in Equation 5.4 are computed, a further analysis is performed based on the total number of diseases. In particular, we are interested in how the average occurrence of states of Equation 5.4 changes when the total number of diseases changes. To facilitate the visualization of results, such average probabilities are grouped per total number of diseases, so that we calculate the *group average* of state s_j for the patients with exactly r diseases, denoted by $g_r(j)$, as follows:

$$g_r(j) = \frac{1}{|D_r|} \sum_{i \in D_r} \overline{\gamma}[i](j) \tag{5.5}$$

where D_r is the set of patients with exactly r diseases. As a result, pairs with number of diseases and group averages are obtained, between which associations, e.g., by the Pearson correlation coefficient, are computed.

5.5 EXPERIMENTAL RESULTS

5.5.1 *Model dimension*

Figure 5.3 shows the model selection scores, which served as a basis for selecting an HMM with 9 states as the suitable model. All the states of the model were associated to fully deterministic emission distributions, such that only one diagnosis variable had a probability equal to 1 in each state, while the other variables had probabilities equal to zero. This means that the property discussed in Section 5.2.1 by which the learned model should emit events with only one active variable (representing the main diagnosis) was met.

5.5.2 *Clusters*

Figure 5.4 shows the learned HMM, where each state is named according to the observable that is active (i.e. the observable that has probability equal to 1). Figure 5.4 shows that three non-unitary clusters were obtained, suggesting that patient visits associated to angina, myocardial infarction and cerebrovascular accident were suitably represented by 2 states each. Intuitively, it is relevant to model a visit to, e.g., angina by means of 2 different states, hence such diagnosis could lead to two different patient courses. As expected, determining which of the two states a visit is associated to depends, e.g., on what is known so far about the patient in terms of past visits.

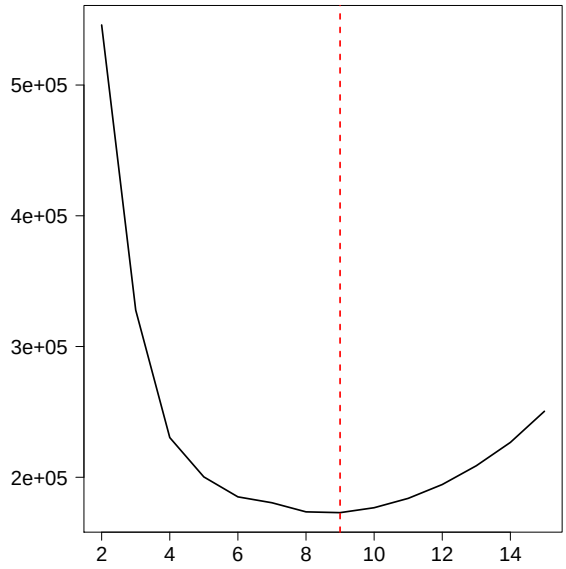


Figure 5.3: Model selection scores. X axis: number of hidden states, Y axis: AIC score. The vertical dashed line indicates the number of states where the AIC was minimal.

5.5.3 *Transition patterns*

Based on the state transitions of Figure 5.4, there is clearly a state in each cluster that will very likely be involved in a self-transition. These states are CVA6, Angina7 and MI3. Such states associate, therefore, to internal patterns in the form of internal recurrent patterns.

The HMM of Figure 5.4 suggests external patterns as well. In particular, angina seems to be a central event in this model: when moving from either the CVA cluster or the MI cluster, it is likely that this transition will reach the Angina cluster (in particular, the Angina5 state). Once in the Angina cluster, a transition to the other clusters is also possible, with probability larger than 0.05. Hence, such external patterns can be thought of as external feedback patterns.

5.5.4 *Clinical interpretation of clusters*

The average probabilities defined in Equation 5.4 are summarized by histograms in Figure 5.5. Each bar corresponds to the number of patients in which a state s_j achieved some average probability. For example, the first bar of CVA2 state means that in around 30,000 patients CVA2 had an average probability between 0 and 11.1%, while for CVA6 the same mean probability was achieved in around 22,500 patients. The histograms allows one to conclude that the CVA6 state was more likely than CVA2 in most patients.

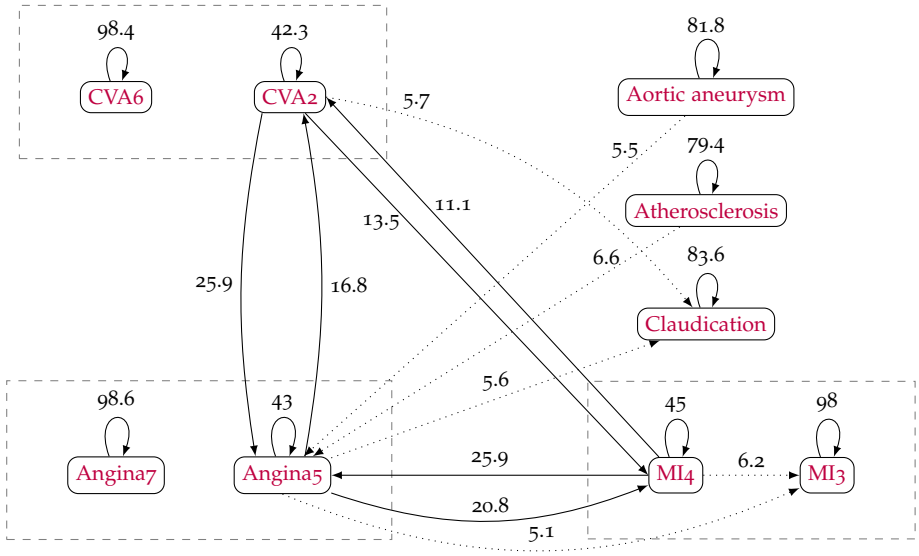
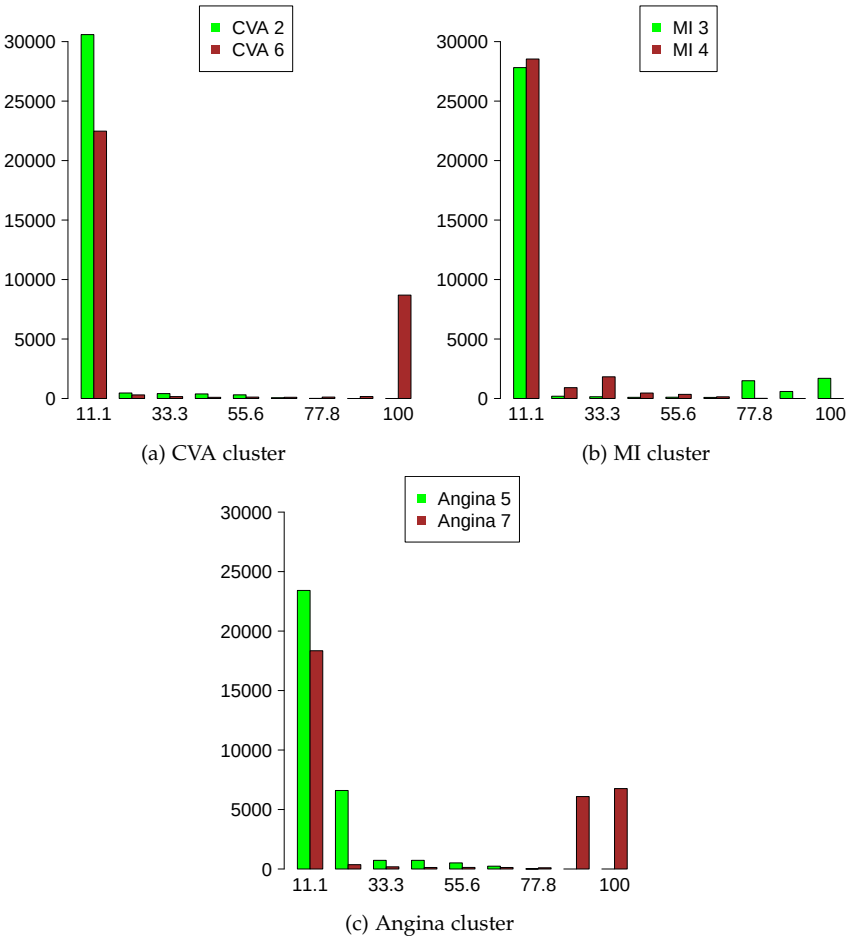


Figure 5.4: Clusters of hidden states, denoted by dashed rectangles. Arcs denote state transitions, with labels indicating probability (in %). For the sake of visualization, transitions with probability between 5 and 10% are shown by dotted lines, and only transitions with probability greater than or equal to 5% are shown.

In general, the histograms of Figure 5.5 suggest that within each cluster there are states that are substantially more prevalent than others, and such separation is more or less uniform depending on the cluster. In general, recurrent-pattern states were more likely than the non-recurrent pattern states, which might suggest that patients likely had several visits due to the same diagnosis before a diagnosis associated to a different comorbidity was registered.

For the second analysis described in Section 5.4.4, Figure 5.6 shows the total number of diseases in patients against the group probabilities. Visual inspection shows that up to 50 diagnoses the trend is substantially more stable than that of all the groups. As around 97% of the patients had at most 50 distinct diagnoses, we will focus on such groups for obtaining a better understanding of the general trend.

Figure 5.6 suggests that, in general, the states of clusters are correlated to the number of diseases in different ways. For the CVA case, patients with only a few diseases are more likely in state CVA6 (internal patterns) rather than CVA2 (external patterns). However, as the number of diseases increases, the chances to be in CVA6 decreases while the chances to be in CVA2 increases, although such trends occur at different paces. Analogously, for an MI event, it is likely the patient will be in state MI3 (internal patterns) if the patient has involves only a few diseases, but a probability decrease is expected for when more diseases are involved. On the other hand, not much can be said about MI4, as the correlation is very low. Intuitively, one would indeed expect that patients with more diseases



(a) CVA cluster

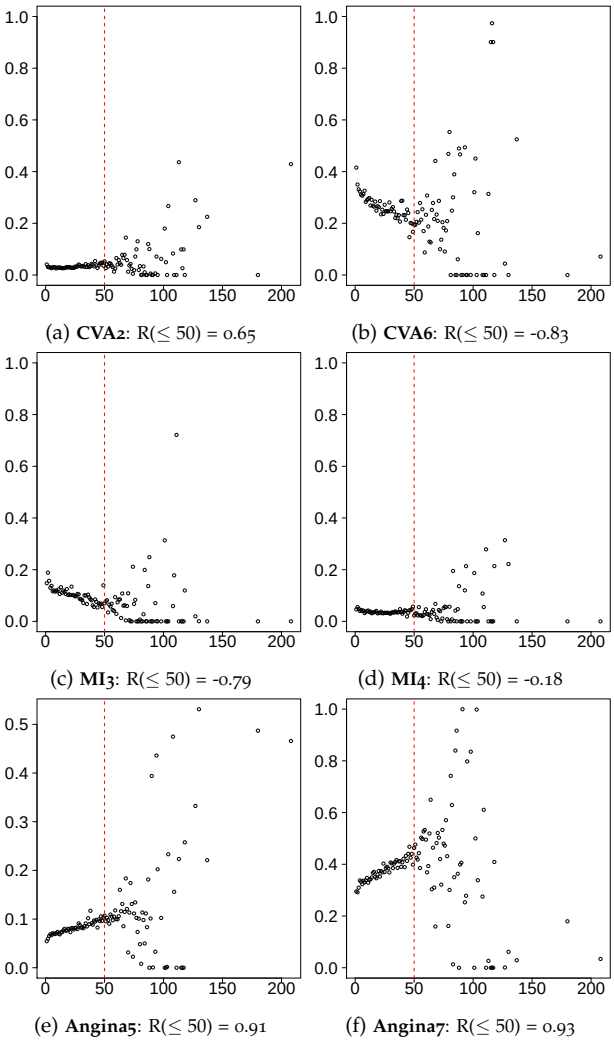
(b) MI cluster

(c) Angina cluster

Figure 5.5: Histograms of average probabilities of states (in %). X axis: average probability of state s_j in the ith patient, i.e. the values $\overline{\gamma}[i](j)$ defined in Equation 5.4. Y axis: number of patients. For example, the first green bar in (a) means that in around 30,000 patients the state CVA2 had an average probability between 0 and 11.1%.

will be related to more transitions between the clusters, which helps explain the observed trends of the CVA and MI clusters.

As opposed to the previous clusters, Figure 5.6 suggests that the dynamics of the Angina cluster has a less straightforward association to the number of diseases. In this cluster, both of its states become more prevalent as the number of diseases increases (up to 50), which might suggest the increasing importance of angina by acting as a proxy for the comorbidities considered in this case study, as well as for other chronic and non-chronic diagnoses not explicitly considered but included in the total number of diseases.



(a) **CVA2**: R($\leq$ 50) = 0.65 (b) **CVA6**: R($\leq$ 50) = -0.83

(c) **MI3**: R($\leq$ 50) = -0.79 (d) **MI4**: R($\leq$ 50) = -0.18

(e) **Angina5**: R($\leq$ 50) = 0.91 (f) **Angina7**: R($\leq$ 50) = 0.93

Figure 5.6: Association of cluster states to clinical outcome (total number of distinct diagnoses). X axis: number of distinct diagnoses, Y axis: group averages $g_r(j)$ (Equation 5.5). The vertical line is drawn at $X = 50$. R indicates the Pearson coefficient, calculated considering only the groups with at most 50 diagnoses (which amounts to 97% of all the patients).

5.5.5 *Are the clusters needed? A comparison to Markov chains*

The need for the clusters learned in the HMM can be assessed by comparing the model fit of the HMM with that of a Markov chain. The state space of such MC is **X**, i.e., the six comorbidities listed in Section 5.4.1, hence learning this MC amounts to estimating the initial and transition probabilities involving the variables in **X**. This comparison can illustrate whether the multiple states

associated to a given comorbidity (in this chapter, the multiple states of CVA, MI and Angina) are indeed necessary for delivering a better model.

Table 5.2 shows the AIC scores computed for the 9-state HMM and for the MC, which indicates a superior model fit for the HMM. Besides such advantage, with the MC it is no longer possible to identify that the occurrence of a certain event such as angina, can be correlated to different patient characteristics (we used in this chapter the total amount of diseases, but other medical outcomes could be devised as well).

Model	**State clusters**	**AIC**
9-state HMM	3 clusters	172,942.8
Markov chain	No clusters	185,013.5

Table 5.2: AIC scores of the HMM and the Markov chain learned from the health care data. The smaller the AIC, the better the model fit is.

5.6 RELATED WORK

The notion of clustering states in hidden Markov models has not been investigated so far to the best of our knowledge. A related approach is clustering applied to timed automata [82, 180], where state sequences are clustered based on their distance by means of hierarchical clustering methods. Based on Bayesian HMMs that use topic modeling, clustering of patient journeys has been proposed [91], which uses the full set of events associated to unstable angina. In contrast, in our case the clusters are determined based on the states, which shifts the focus towards the dynamics that involve states within clusters. Despite their differences, our methods and those from the literature share the goal of moving towards explainable artificial intelligence [80, 114], as we aimed not only to obtain a model with suitable fit, but also to understand more about the patient situation by looking at the structure of the HMM. An example in our case is the deterministic emissions, which can facilitate interpreting models like HMMs to a great extent, at the same time obeying constraints of the multimorbidity problem.

In the context of electronic health records of multimorbidity, a cohort of the NIVEL data used in this chapter had been used for learning graphical models based on Bayesian networks, in static [106] and temporal [107] contexts. In those cases, however, the goal was to model differences in practices, hospitals, or regions, without taking into account latent variables.

5.7 CONCLUSIONS

In this chapter we proposed a modeling methodology for health care data from EHRs. Due to the fine-grained nature of such event data, we used HMMs for capturing latent information that is not directly measured. A first step towards

capturing clusters of latent states was taken, which are states associated to the same emission distribution. In the context of EHR data, the states of a cluster are associated to the same diagnosis code. The states of a cluster can, however, be associated to very different transitioning patterns. Based on this, we defined the notion of transition patterns.

We illustrated the proposed ideas by means of a case study with data from atherosclerosis patients collected by Dutch general practitioners. The learned HMM had 9 states, in which clusters involving angina, myocardial infarction and cerebrovascular accident were identified. This suggests that these diagnoses are too complex to be managed by a single latent state, hence a model with better fit was obtained when such diagnoses were allowed to be represented by multiple states (or roles), as we did with the obtained HMMs.

Suggestions for future work include a complementary analysis to the correlations computed between average state probabilities and the total number of diseases. Instead of computing separate correlations, one could consider regression models to predict the average probabilities for different number of diseases and states. In terms of model class, we also would like to investigate the effect of adding medication and lab exams, which are available to some patients in the NIVEL data. These could be added as model inputs (i.e. covariates), which would allow to capture switching regimes for the transitions.

Further research might also benefit from a more formal definition of clusters of hidden states allowing one to capture more general transition patterns. This could make the patterns more explainable. One could also add criteria to help decide which states are part of a cluster in a more general way, which could be of interest if the emissions are not fully deterministic (e.g. when there is a second diagnosis available in the data).

6

PARTITIONED DYNAMIC BAYESIAN NETWORKS

When modeling the dynamics of real-world processes, the model properties are often assumed to be constant over time, resulting in a so-called time-homogeneous process. This might be justified, e.g., by scarce amounts of data available. While this reduces the number of parameters to be learned from data, the specificities of the underlying process are to some degree lost in the obtained models. In this chapter, we propose partitioned dynamic Bayesian networks *for capturing distribution regime changes, benefiting from an intuitive and compact representation with the solid theoretical foundation of Bayesian network models. In order to balance specificity and simplicity in real-world scenarios, we propose a heuristic algorithm to search and learn such models taking into account the preference for less complex models. Experiments are performed based on simulated data to evaluate how well the proposed method is able to recover the original distributions, for different assumptions regarding the data generating mechanism. Finally, we consider a study case based on psychotic depression complementary to that of Chapter 4 to evaluate the goodness-of-fit and insight that partitioned dynamic Bayesian networks can provide to a real-world problem.*

6.1 INTRODUCTION

Understanding the evolution of disease processes lies at the heart of clinical medicine as insights into how effective a particular treatment is able to cure a disease are based on this. Not surprisingly, most textbooks on clinical medicine and pathology contain extensive descriptions of how a disease progresses and likely reacts to particular treatments in the course of time. Yet, there has been very little research where these qualitative descriptions have been substantiated in a detailed, quantitative way. In research, the temporal dimension is usually only explored by describing the outcome of treatment after some time. One of the problems faced by researchers who wish to obtain such insight is the relatively small size of clinical datasets. Often, data concerns something from a hundred to a few hundreds of patients. However, the wish to develop a temporal model usually increases the demands for data, and as a consequence various simplifying assumptions have to be made.

One solution that is usually considered in clinical problems is to build a model that covers the entire time span without distinguishing any of its time points [29,

74, 101, 146, 157]. Therefore, the model has the same properties for every time point, as modeled by the well-known first-order homogeneous Markov chains [52]. A generalization of Markov chains to multivariate problems are the dynamic Bayesian networks [104, 136], which have been applied to a number of real-world domains, such as medicine [38, 86, 132, 153] and bioinformatics [54, 109, 145]. Such probabilistic graphical models allow to reason about the interactions of features of interest in an intuitive, temporal and compact fashion, while having a sound basis in probability theory. This will yield more robust models, making the use of these models attractive when dealing with small datasets. However, while DBNs solve the robustness problem, they introduce an undesirable effect: there is no distribution specificity as a function of time. Hence, one will never learn the details of the underlying process as was the aim in the first place.

It is known that in many clinical situations the dependences between symptoms and signs might change over time, as in the case of intervention studies where different sets of correlations are expected to occur in the course of time, due to the nature of this kind of study. Hence, a temporal graphical model that is allowed to vary in structure and probability distribution as a function of time would capture these complex dynamics, providing a potentially better model fit and more insight that really helps in understanding the underlying process.

Although the notion of non-homogeneous models (a shorthand for *non-homogeneous time models*) is certainly not new, it is often the case that such models employ a number of approximations, for example due to properties of the targeted applications. Typically, non-homogeneous PGMs have been focused on biological processes, where regime shifts are assumed to be smooth [79, 109, 145]. These assumptions might, however, not be natural for other processes, where the variety of eligibility criteria and unexpected patient response to drugs can make the distribution regimes over time vary widely. Thus, a systematic algorithm that finds the appropriate cut-off points to obtain new specific models, taking into account the scarcity of data and the wish to obtain a robust model, is needed. To the best of our knowledge, this idea has never before been explored in learning Bayesian network-based models from data.

In this chapter, we first introduce *partitioned dynamic Bayesian networks* (PDBNs, for short), which allow to express a process as a collection of DBNs. PDBNs make few assumptions regarding the process, the main one being the fact that the process duration is partitioned in the same way for every observable variable involved. Then, we propose a heuristic procedure to explore the space of PDBNs, taking into account the balance between specificity and simplicity. The approach starts with a homogeneous model, and incrementally replaces parts of it by sub-models that are valid for specific time periods. The increase of complexity is allowed if there is a two-part split of one of the current sub-models that is able to improve model fit over a training and test setting.

In order to demonstrate the applicability of the proposed model and heuristic method, an extensive set of simulations and real-world-based experiments are carried out. In simulations we evaluate whether the heuristic algorithm is able to recover adequate models in terms of statistical distance to the data generating

model, be it a homogeneous or a non-homogeneous model. We also aim to evaluate experimentally the behavior of the heuristic in the case of small datasets. Additionally, we consider a study case on psychotic depression data, and evaluate the homogeneous and non-homogeneous models learned from this data. Based on the obtained models, research questions of clinical relevance are formulated regarding the prediction of symptom association over time.

The remainder of this chapter is organized as follows. Section 6.2 describes related literature on homogeneous and non-homogeneous dynamic Bayesian networks in clinical and biological domains. Partitioned DBNs and the heuristic procedure to learn PDBNs are presented in Section 6.3. Simulations to evaluate the learning procedure are discussed in Section 6.4, while the models learned from psychiatry data are discussed in Section 6.5. Clinically-oriented discussions based on the psychiatry models are provided in Section 6.6, and lastly Section 6.7 gives the conclusions and suggestions for future research.

6.2 RELATED WORK

There has been quite some research on the application of Bayesian network models to the clinical domain. To a lesser extent, models that take time into account, such as dynamic Bayesian networks, have been considered in the past. Relevant research include obtaining problem insight by analyzing the structure and parameters of a DBN, and the use of DBN models for specific tasks such as diagnosis and prognosis. For example, the learned structure of DBNs has been explored for finding correlations among different brain regions in several disorders, such as schizophrenia [101] and Alzheimer's disease [29]. These results have been used to confirm known correlations as well as to reveal new ones. Furthermore, the sensitivity of the influence of parameter variation in DBNs has been investigated in the context of ventilator-associated pneumonia [37].

Another aspect of DBNs explored in the clinical domain is the predictive ability for several tasks, e.g. diagnosis [38, 146] and prognosis [74]. An advantage of modeling stochastic processes using models as DBNs lies in the capability of producing updated predictions as new observations become available while the process evolves. This can be achieved by taking into account some form of patient history, producing potentially more accurate predictions. Real cases have shown the benefits of this type of multiple prediction, e.g. to diagnose ventilator-associated pneumonia [38]. The application of DBNs and similar models in clinical domains has been compared to similar formalisms in a recent survey [132].

Although DBNs have been reasonably studied for their capability to deal with clinical problems, this is not the case for more flexible models, e.g. when the time-homogeneity assumption is rejected. These models address mainly the analysis of change in structure at individual time points, in the scope of a specific disease process [171]. On the other hand, more sophisticated models have been developed in other fields, mainly biological processes [54, 79, 109, 145]. These models are constructed based on assumptions justified by domain knowledge;

for example, in some biological processes the intensity of interactions change over time, but no interaction is created or destroyed [79].

The aforementioned non-homogeneous models assume a set of assumptions or use a specific learning methodology, which we summarize as follows. Firstly, additional restrictions are usually imposed to the model structure, ranging from constrained intra-temporal interactions [109, 145] to completely fixed structure with flexibility on the parameter space only [79]. A second assumption is that regime switch in the process occurs in a smooth fashion. Finally, in many biological-oriented networks the learning approach is based on sampling strategies [54, 79, 109, 145], which can depend on additional assumptions in order to be feasible. As we show further in the chapter, these assumptions will not be considered for the development of PDBNs. Other approaches include, e.g., DBN models with hidden variables to control the dependence structure, which has been applied to engineering problems [170].

Clearly, clinical problems are potentially prone to exhibit a temporal behavior that may be different from the biological processes studied so far. To illustrate this, consider the case of intervention studies, where specific criteria exist to define eligible patients. Imposing the previous assumptions on the manner by which pieces of the process evolve can forbid capturing the temporal dynamics accurately. Therefore, there is a need to define and construct models of non-homogeneous time in a systematic manner, which will be able to reveal more about the underlying structure of processes in clinical domains.

6.3 PARTITIONED DYNAMIC BAYESIAN NETWORKS

Models of non-homogeneous time can be defined by a set of transition distributions that should hold at specific intervals of the considered time series. In this work, the central idea lies in making the dependence on time by partitioning the time series duration and associating each part to a homogeneous model, i.e. a DBN valid within a sub-range of the time series. We refer to this class of models as *partitioned dynamic Bayesian networks*. We proceed in the following towards a formalization of PDBNs, its associated concepts, and lastly a procedure to learn PDBNs by exploring the search space heuristically.

6.3.1 *Model specification*

Definition 6.1 (Time partition)**.** *A time partition of a set of integers $\{0, \ldots, T\}$ is a set of integers $\{t_1, \ldots, t_k\}$, where $t_1 > 0$, $t_k = T$, and $t_i < t_{i+1}$ for $1 \leq i < k$. Each t_i, with $i > 1$, defines a set $\{1 + t_{i-1}, \ldots, t_i\}$, and t_1 defines the set $\{0, \ldots, t_1\}$.*

We say that each element of the time partition is a cut (a shorthand for cut-off) and we say that such time partition has k cuts.

The aim of Definition 6.1 is to split a time series horizon into a partition of indices. For example, given a time series indexed by the time points $\{0, \ldots, 7\}$, the time partition $\{2, 7\}$ has 2 cuts and splits the time series as follows: $\{0, 1, 2\}$,

and $\{3,4,5,6,7\}$. This definition is useful for defining non-homogeneous models as follows.

Definition 6.2 (Partitioned dynamic Bayesian network)**.** *Consider a time partition with k cuts of the integers $\{0,\ldots,T\}$, where the ith cut is associated to a conditional Bayesian network $\mathcal{B}_i$ over $\mathbf{X}^{(t+1)}$ conditioned on $\mathbf{X}^{(t)}$, $t \geq 0$. A partitioned dynamic Bayesian network with k cuts, denoted by PDBN-k, is a dynamic system $(\mathcal{B}_0,\ldots,\mathcal{B}_k)$ over $\mathbf{X}$ where:*

- *$\mathcal{B}_0 = (\mathcal{G}_0, P_0)$ is a Bayesian network over the variables $\mathbf{X}^{(0)}$ called initial network.*
- *$\mathcal{B}_i = (\mathcal{G}_i, P_i)$, $i > 0$, is a conditional Bayesian network over the variables $\{\mathbf{X}^{(t)}, \mathbf{X}^{(t+1)}\}$ called the ith transition network. The transition model $\mathcal{B}_i$ is associated to the ith cut of the time partition.*

We use the term *distribution cut* to denote a cut in the context of a PDBN. The joint distribution of an unrolled PDBN can be obtained by unrolling the transition models over the time points each transition model is associated to. This is as follows: the structure and parameters of all the nodes at time $t = 0$ come from the initial model $\mathcal{B}_0$, while the structure and parameters for any node $X_i^{(t)}$, where $t > 0$, come from the transition model whose cut includes t, i.e., the B_i such that $t \in \{1 + t_{i-1},\ldots,t_i\}$. Therefore, the joint distribution of an unrolled PDBN with k cuts $\{t_1,\ldots,t_k\}$ is as follows:

$$\begin{aligned} P(\mathbf{X}^{(0:T)}) = &\prod_{i=1}^{n} P_0(X_i^{(0)} \mid \pi(X_i^{(0)}, \mathcal{B}_0)) \\ &\cdot \prod_{r=1}^{k} \prod_{t=t_{r-1}}^{t_r-1} \prod_{i=1}^{n} P_r(X_i^{(t+1)} \mid \pi(X_i^{(t+1)}, \mathcal{B}_r)) \end{aligned} \tag{6.1}$$

where $t_0 = 0$ and P_r refers to the CPTs pertaining to the transition model $\mathcal{B}_r$. Note that the parent set of each X_i depends on $\mathcal{B}_r$ as denoted by $\pi(X_i, \mathcal{B}_r)$.

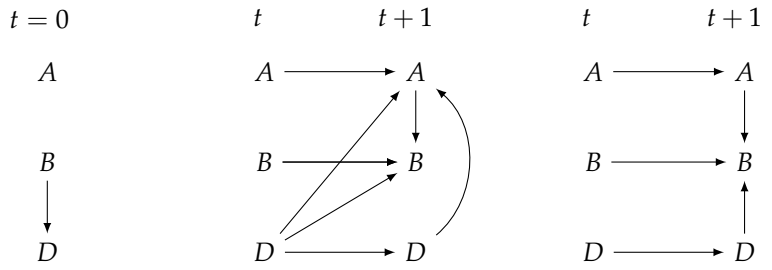
It follows from the previous definitions that a DBN is a PDBN with a single cut $\{T\}$, hence, a DBN is a PDBN-1.

Example 6.1. *Consider again the situation of Example 2.2, where two symptoms A and B and a drug quantity D are measured per patient on a regular basis. We define a PDBN-2 for this problem consisting of two cuts $\{2,7\}$ whose initial structure and transition structures are shown on Fig. 6.1. Each cut of the PDBN is associated to a conditional BN as follows: $\mathcal{B}_1$ holds for the time points $\{0,1,2\}$, while $\mathcal{B}_2$ holds for the time points $\{3,4,5,6,7\}$.*

Unrolling this PDBN-2 for the process duration yields the joint

$$\begin{aligned} P(\mathbf{X}^{(0:7)}) = &\prod_{i} P_0(X_i^{(0)} \mid \pi(X_i^{(0)}, \mathcal{B}_0)) \\ &\cdot \prod_{0 \leq t \leq 1} \prod_{i} P_1(X_i^{(t+1)} \mid \pi(X_i^{(t+1)}, \mathcal{B}_1)) \\ &\cdot \prod_{2 \leq t \leq 6} \prod_{i} P_2(X_i^{(t+1)} \mid \pi(X_i^{(t+1)}, \mathcal{B}_2)) \end{aligned} \tag{6.2}$$

where $X_i \in \mathbf{X}$, $\mathbf{X} = \{A, B, D\}$, *and* P_i *refers to the CPTs pertaining to the transition model* $\mathcal{B}_i$.



(a) Initial model $\mathcal{B}_0$ (b) Transition model $\mathcal{B}_1$ (c) Transition model $\mathcal{B}_2$

Figure 6.1: An example of PDBN-2. $\mathcal{B}_i$ represents the ith transition model (only its structure is shown, parameters are omitted). Nodes on the left and right side occur at t and $t+1$ respectively, except for the initial model.

6.3.2 *A heuristic search procedure*

In this section, we present a heuristic algorithm to build PDBNs in an incremental fashion from a dataset of sequences. As in many clinical studies there is typically a scarcity of data, mainly in terms of number of sequences (e.g. represented by patients), the central idea of the procedure is to prefer less complex models. In order to achieve this, the heuristic assumes that a proper criterion for model selection that prevents overfitting is used, which is naturally dependent on the application domain and characteristics of the data. Hence, when constructing a model, the heuristic iteratively increases the complexity as long as it is beneficial for its score; if adding complexity is not beneficial, the procedure stops adding further complexity. Additionally, the procedure has a hill-climbing behavior by not further exploiting previous less complex solutions that were less promising when analyzed by the algorithm.

6.3.2.1 *Algorithm description*

Taking the aforementioned factors into account, we present a procedure that starting from a DBN follows a sequence of incremental refinements to evolve it into a more specialized model. A refinement corresponds to splitting one of the transition distributions of the current PDBN. At each iteration a new cutting point is added without eliminating the cuts previously found. The procedure is greedy since it does not further explore the branching of solutions that are less interesting at each iteration. It is important to consider a strategy with feasible running time to search over the space of PDBNs, since the number of possible manners in which a discrete time series can be partitioned is potentially large. In order to be flexible, the complexity of the produced models can be controlled, as it is an input parameter of the algorithm.

The heuristic algorithm to learn PDBNs is presented in Algorithm 2. In order to be generic for different scoring criteria used to construct and evaluate PDBNs, we emphasize the search for cut sets instead of PDBNs explicitly. The algorithm starts with the current best cut set as the singleton $C = \{T\}$, which stands for a homogeneous model. Let us denote by s the size of the current cut set, i.e., $s = |C|$. By entering the outer loop (Line 2) the algorithm will first evaluate new cut sets with size $s + 1$, each one consisting of the current C unified with a new cut that does not exist in C (Line 3). After finishing the inner loop, it is verified whether the current iteration has found an improved cut set, i.e., a cut set whose evaluation is better than C. In case positive, C is replaced by the best cut set among those (Lines 5-6). The algorithm continues this incremental construction of cut sets while the current iteration is capable of producing a new cut set with size $(s + 1)$ that is better than the current C and the maximum number of cuts (the input parameter k) is not reached. At the end (Line 8), the heuristic returns the PDBN-k' learned from the best cut set found, where $k' \leq k$.

Algorithm 2 Builds a PDBN

Input: D: a dataset of sequences with length $\{0, \ldots, T\}$;
k: the maximum size of the cut set, $1 \leq k \leq T$.
Output: a PDBN-k', where $k' \leq k$.

1: $C \leftarrow \{T\}$
2: **while** $|C| < k$ **do**
3: For each $c \in \{1, \ldots, T\} - C$, construct a new cut set $C \cup \{c\}$. Denote the new cut sets by $\mathbf{C} = \{C_1, \ldots, C_r\}$.
4: Evaluate each cut set in $\mathbf{C}$ by means of a criterion f.
5: **if** there is a new cut set $C_i \in \mathbf{C}$, where $1 \leq i \leq r$, such that $f(C_i) > f(C)$ **then**
6: Assign to C the C_j that maximizes $\{f(C_1), \ldots, f(C_r)\}$.
7: **else** break the loop.
8: **return** PDBN-k' with cut set C learned from the data D.

6.3.2.2 *Evaluation criterion*

As Algorithm 2 shows, the criterion f abstracts the learning of PDBNs. This is motivated by the fact that choosing a proper evaluation strategy depends on the application and the characteristics of the data, which makes it difficult to set a single criterion that works best for all problems [182]. Generally speaking, a multitude of model selection criteria can be employed to determine how f is concretely implemented; some well-known criteria include cross-validation (e.g. based on model likelihood) and information theory criteria (e.g. the Akaike information criterion and the Bayesian information criterion) [46].

For example, in order to employ the AIC in Algorithm 2, one would first learn a PDBN from the full dataset (i.e. all the sequences, hence a DBN) using the

AIC as scoring function. Then, each sub-DBN associated to new cut sets (Line 3) would be learned based on this score using the corresponding part of the data.

6.3.2.3 *Complexity*

Initially, the cut set maintained by the algorithm is $C = \{T\}$. At the first iteration of the outer loop, new cut sets with size $s + 1$ are built, consisting of C plus a new element; there are $T - 1$ manners to make this inclusion. At the second iteration, there are $T - 2$ possible cut sets to be constructed, and so on, until the last iteration, in which there is only one cut to be inserted in the current C. Thus, the total number of cut sets constructed by the heuristic is in $\mathcal{O}(T^2)$, considering the worst case.

The dominant part of the heuristic's total cost corresponds to learning models. In the case of learning DBNs, the input can be seen as a transition dataset $(\mathbf{X}, \mathbf{X}')$, consisting of all the data $(\mathbf{X}^{(i)}, \mathbf{X}^{(i+1)})$, $i = 0, \ldots, T - 1$, merged. Note that this construction is sound since the model is time-homogeneous. If the original dataset D consists of m sequences (each of length $T + 1$), this merged dataset will consist of mT short sequences (each of length 2). Thus, abstracting the cost of learning a DBN by means of a cost function g will lead to a cost of $\mathcal{O}(g(mT))$ for learning a DBN.

The case of learning PDBNs-k, $k > 1$, can be seen as learning k sub-DBNs made of potentially different number of sequences, as dictated by the cut set of the PDBN. Note that when the number of cuts is maximal, it implies learning T sub-DBNs, each one from a transition dataset $(\mathbf{X}^{(i)}, \mathbf{X}^{(i+1)})$ consisting of m sequences, each with length 2. As each of these sub-DBNs would cost $g(m)$, learning such PDBN would require $\mathcal{O}(Tg(m)))$.

6.4 EMPIRICAL EVALUATION VIA SIMULATIONS

6.4.1 *Simulation parameters*

In this section experiments based on simulated data are presented for a general assessment of the proposed method for learning PDBNs. Time series with varying length and number of sequences were generated, resulting in diversified datasets. We considered the number of features as $n \in \{2, 6, 10, 14, 18\}$, and defined that each time series is composed by sequences with length of 10 or 30 time points. Hence, the unrolled models used in simulations have between 20 and 540 random variables in total. For each n and time series length, datasets were randomly generated containing different number of sequences, denoted by $d \in \{100, 500, 2000, 5000\}$. Thus, the simulation cases allow for a reasonable evaluation in terms of different feature spaces and dataset sizes.

For each simulation scenario, a random DBN or PDBN-k was constructed, consisting of n binary features per instant t. Structurally, a random PDBN-k consists of k random sub-DBNs, where the graphical structure of each random sub-DBN was uniformly generated at random [122], and distribution parameters

determined randomly as well (no noise was introduced in the model's parameters). Hence, each node of an unrolled PDBN assumes a Bernoulli distribution. Given a random PDBN-k and a random cut set of length k, whose last cut corresponds to the length of the sequences that are to be sampled from the model, four distinct datasets were constructed, one for each value of d. In other words, a common underlying model was used for each group of simulations since the experiments also aims at studying the effect on the heuristic's capabilities over different quantities of data.

Each dataset was generated from either a random DBN or a random PDBN. The initial aim is to verify experimentally whether the construction algorithm is able to learn the adequate class of model with respect to the reference model (a random DBN or PDBN) used to simulate data. Moreover, the cuts of the learned models are compared to the cuts of the reference models, where we use the following notation:

- If the cuts of the reference and learned models are equal, we write '$=$'.
- If the cuts of the learned model include all the cuts of the reference model, we write '$\subseteq +a$', where a denotes the number of additional cuts included by the learned model.
- If none of these criteria is met, we write '$\not\subseteq$'.

Although this notation is useful to perform a structural comparison in terms of the number and position of distribution cuts, they do not provide information about the distance between the probability distributions of two models. To this end, the Kullback-Leibler (KL, for short) divergence [46] between the marginal distribution of each feature $X_i^{(t)}$ was considered, which indicates the amount of additional information one needs to codify samples from one distribution using another distribution. The KL divergence over the entire joint distribution is computationally prohibitive for most of the simulations covered in this section, therefore we compute the KL divergence over marginal distributions as follows:

$$\sum_{i=1}^{n}\sum_{t=0}^{T} \mathrm{KL}(P(X_i^{(t)}) \mid\mid Q(X_i^{(t)})) = \sum_{i=1}^{n}\sum_{t=0}^{T}\sum_{X_i} P(X_i^{(t)}) \log \frac{P(X_i^{(t)})}{Q(X_i^{(t)})} \tag{6.3}$$

where $Q(X_i^{(t)}) = 0$ implies $P(X_i^{(t)}) = 0$. Equation 6.3 corresponds to the sum of the divergences between the marginal distributions P and Q, in this case a reference distribution and a learned distribution respectively. As with the standard KL divergence, the quantity of Equation 6.3 should be minimized.

6.4.2 *Learning and evaluating PDBNs*

In order to learn a PDBN with k cuts, k homogeneous models are learned using the corresponding portions of the training data according to its cut set, where each sub-DBN is learned separately. As it happens with Bayesian-network

learning, typically search-and-score and constraint-based methods are used for learning the structure and parameters of each sub-DBN (see Section 2.4.2). In the experiments reported in this chapter, the AIC score (see Equation 5.2) is employed for evaluating each sub-DBN, which yields a score proportional to the likelihood of the model and a penalization term for the complexity.

In order to select a suitable number of cuts, we implemented the evaluation criterion of Algorithm 2 by means of a 10-fold cross-validation. Cross-validation minimizes the effect of overfitting (see Section 2.6.3); we describe the procedure in detail in the following. Let $C_i = \{t_1, \ldots, t_k\}$ be a cut set of a time series over $\{0, \ldots, T\}$; in the context of Algorithm 2, C_i corresponds to a new cut set that is built in Line 3. For each cross-validation fold, the training data is used to learn a PDBN-k with cut set C_i, while the test data is used to compute the log-likelihood of such PDBN-k. After processing all the folds, the mean of the log-likelihoods is taken, which represents the evaluation value of the PDBN-k with cut set C_i, as indicated in Algorithm 2 by $f(C_i)$. When deciding between two cut sets (e.g. as in Line 5), the algorithm chooses the one having the higher mean log-likelihood.

After leaving the outer loop of Algorithm 2, the heuristic search is finished and the best cut set is known. Finally, a PDBN-k with such cut set is learned using the full dataset, i.e. training and test data. Such PDBN-k corresponds to the output of the procedure.

6.4.3 *Results and discussion*

The results of simulations with data generated from DBN, PDBN-2 and PDBN-3 models are shown in Tables 6.1, 6.2 and 6.3 respectively. Note that a DBN was learned on every case to serve as a baseline method, specially when simulating data from non-DBNs; the performance of the learned DBNs are indicated on the sixth column of the tables. Table 6.1 shows that the models learned by the heuristic based on DBN data have structural partitioning in accordance with the reference models on most cases, indicating that the heuristic was capable of retrieving the adequate type of model. When the returned models were not a DBN, they were mostly only slightly more complex ones (i.e. PDBNs-2). Interestingly, the KL divergence between the learned PDBNs and the respective reference models are comparable to the divergence of the learned DBNs, i.e. although consisting of additional transition distributions, the learned PDBNs captured the reference distribution as well as the learned DBNs did.

The models returned by the heuristic based on data produced by PDBNs-2 and PDBNs-3 (Tables 6.2 and 6.3) support analogous points discussed just before. Furthermore, these tables show that the KL divergences of the PDBNs learned heuristically were substantially lower than those of the learned DBNs, i.e. the former are closer to the reference ones. This fact was more prominent when the length of the time series was increased to 30. Intuitively, DBNs capture the average behavior of the distribution underlying data; if most of the transitions were originated from a single distribution, then the few remaining ones will tend to have less impact on the distribution learned by the DBN. On the PDBN-2

n	d	Learned Model	Cut Sets (R; L)	Cut Diff.	KL(L DBN)	KL(L)
Time series length = 10						
2	100	DBN	(9)	=	0.04	0.04
2	500	DBN	(9)	=	0.01	0.01
2	2000	DBN	(9)	=	0	0
2	5000	PDBN-2	(9); (7,9)	$\subseteq$ +1	0	0
6	100	DBN	(9)	=	0.17	0.17
6	500	DBN	(9)	=	0.04	0.04
6	2000	DBN	(9)	=	0.01	0.01
6	5000	DBN	(9)	=	0.01	0.01
10	100	DBN	(9)	=	0.24	0.24
10	500	DBN	(9)	=	0.09	0.09
10	2000	DBN	(9)	=	0.02	0.02
10	5000	DBN	(9)	=	0.02	0.02
14	100	DBN	(9)	=	0.38	0.38
14	500	DBN	(9)	=	0.07	0.07
14	2000	DBN	(9)	=	0.03	0.03
14	5000	DBN	(9)	=	0.02	0.02
18	100	DBN	(9)	=	0.23	0.23
18	500	DBN	(9)	=	0.07	0.07
18	2000	DBN	(9)	=	0.03	0.03
18	5000	DBN	(9)	=	0.02	0.02
Time series length = 30						
2	100	DBN	(29)	=	0.01	0.01
2	500	DBN	(29)	=	0.01	0.01
2	2000	DBN	(29)	=	0	0
2	5000	PDBN-2	(29); (1,29)	$\subseteq$ +1	0.01	0.01
6	100	DBN	(29)	=	0.16	0.16
6	500	DBN	(29)	=	0.03	0.03
6	2000	DBN	(29)	=	0.02	0.02
6	5000	DBN	(29)	=	0.02	0.02
10	100	DBN	(29)	=	0.13	0.13
10	500	DBN	(29)	=	0.04	0.04
10	2000	DBN	(29)	=	0.03	0.03
10	5000	DBN	(29)	=	0.02	0.02
14	100	DBN	(29)	=	0.26	0.26
14	500	DBN	(29)	=	0.07	0.07
14	2000	DBN	(29)	=	0.04	0.04
14	5000	DBN	(29)	=	0.04	0.04
18	100	DBN	(29)	=	0.3	0.3
18	500	DBN	(29)	=	0.08	0.08
18	2000	DBN	(29)	=	0.05	0.05
18	5000	DBN	(29)	=	0.04	0.04

Table 6.1: Simulations with *data generated from DBNs*, where n and d denote the number of features and the number of sequences respectively. **R** = reference model, **L** = learned model (heuristic), **KL (M)** = KL divergence between model M and the reference model, **L DBN** = learned DBN.

and PDBN-3 cases where the first cut was situated around half of the sequence duration, there were at least two different transition patterns, which tends to make DBNs less representative of each individual transition.

Overall, it is worth noting that the cases where the heuristic procedure was not capable of constructing models with the same structural partition of transitions as the reference models do have some particularities. Namely, these cases contain just a few features (mostly $n = 2$) or have few sequences. Despite not returning the exact type of model, the KL divergences of these PDBNs were noticeably

smaller than the divergences of the learned DBNs, suggesting that the heuristic made mistakes with low impact nonetheless.

n	d	Learned Model	Cut Sets (R; L)	Cut Diff.	KL(L DBN)	KL(L)
Time series length = 10						
2	100	PDBN-4	(1,9); (1,2,4,9)	$\subseteq$ +2	0.18	**0.08***
2	500	PDBN-2	(1,9)	=	0.16	**0.02***
2	2000	PDBN-4	(1,9); (1,5,7,9)	$\subseteq$ +2	0.19	**0.01***
2	5000	PDBN-4	(1,9); (1,5,8,9)	$\subseteq$ +2	0.19	**0.01***
6	100	PDBN-2	(6,9)	=	1.88	**0.14***
6	500	PDBN-2	(6,9)	=	1.81	**0.03***
6	2000	PDBN-2	(6,9)	=	1.76	**0.01***
6	5000	PDBN-2	(6,9)	=	1.76	**0.01***
10	100	DBN	(8,9); (9)	$\not\subseteq$	1.06	1.06
10	500	PDBN-2	(8,9)	=	0.96	**0.05***
10	2000	PDBN-2	(8,9)	=	0.95	**0.02***
10	5000	PDBN-2	(8,9)	=	0.95	**0.01***
14	100	PDBN-2	(3,9)	=	3.07	**0.37***
14	500	PDBN-2	(3,9)	=	2.68	**0.1***
14	2000	PDBN-2	(3,9)	=	2.39	**0.03***
14	5000	PDBN-2	(3,9)	=	2.35	**0.02***
18	100	DBN	(1,9); (9)	$\not\subseteq$	1.57	1.57
18	500	PDBN-2	(1,9)	=	1.04	**0.09***
18	2000	PDBN-2	(1,9)	=	0.93	**0.02***
18	5000	PDBN-2	(1,9)	=	0.78	**0.02***
Time series length = 30						
2	100	PDBN-2	(15,29)	=	5.05	**0.09***
2	500	PDBN-2	(15,29)	=	5.05	**0.02***
2	2000	PDBN-7	(15,29); (2,6,15,20,26,28,29)	$\subseteq$ +5	5.07	**0.02***
2	5000	PDBN-4	(15,29); (10,15,25,29)	$\subseteq$ +2	5.08	**0.01***
6	100	PDBN-2	(18,29)	=	15.8	**0.12***
6	500	PDBN-2	(18,29)	=	15.7	**0.04***
6	2000	PDBN-2	(18,29)	=	15.76	**0.02***
6	5000	PDBN-2	(18,29)	=	15.95	**0.02***
10	100	PDBN-2	(20,29)	=	7.24	**0.26***
10	500	PDBN-2	(20,29)	=	7.25	**0.12***
10	2000	PDBN-2	(20,29)	=	7.2	**0.06***
10	5000	PDBN-2	(20,29)	=	7.13	**0.03***
14	100	PDBN-2	(21,29)	=	9.29	**0.35***
14	500	PDBN-2	(21,29)	=	9.09	**0.09***
14	2000	PDBN-2	(21,29)	=	9.09	**0.06***
14	5000	PDBN-2	(21,29)	=	9.02	**0.04***
18	100	PDBN-2	(17,29)	=	13.02	**0.34***
18	500	PDBN-2	(17,29)	=	12.82	**0.1***
18	2000	PDBN-2	(17,29)	=	12.57	**0.06***
18	5000	PDBN-2	(17,29)	=	12.64	**0.05***

Table 6.2: Simulations with *data generated from PDBN-2 models*. The best KL divergence values are given in bold face and followed by an asterisk.

A summary of the results presented in Tables 6.1, 6.2 and 6.3 is given in Table 6.4. Each row of the table aggregates simulations of DBNs, PDBNs-2 and PDBNs-3 according to the number of features and sequence length.

6.4.4 *Small datasets*

In the final analysis based on simulations, we focus on the small datasets. The simulations suggest that the models learned by the heuristic from the smallest

n	d	Learned Model	Cut Sets (R; L)	Cut Diff.	KL(L DBN)	KL(L)
Time series length = 10						
2	100	PDBN-2	(1,6,9); (6,9)	$\not\subseteq$	2.73	**0.26***
2	500	PDBN-4	(1,6,9); (1,2,6,9)	$\subseteq$ +1	2.8	**0.02***
2	2000	PDBN-3	(1,6,9)	=	2.79	**0.01***
2	5000	PDBN-4	(1,6,9); (1,4,6,9)	$\subseteq$ +1	2.78	**0***
6	100	PDBN-3	(2,6,9)	=	3.37	**0.25***
6	500	PDBN-3	(2,6,9)	=	3.09	**0.04***
6	2000	PDBN-3	(2,6,9)	=	2.91	**0.02***
6	5000	PDBN-3	(2,6,9)	=	2.94	**0.01***
10	100	PDBN-2	(6,8,9); (6,9)	$\not\subseteq$	2.99	**1.83***
10	500	PDBN-3	(6,8,9)	=	2.85	**0.07***
10	2000	PDBN-3	(6,8,9)	=	2.8	**0.02***
10	5000	PDBN-3	(6,8,9)	=	2.78	**0.02***
14	100	PDBN-2	(2,3,9); (3,9)	$\not\subseteq$	6.5	**1.61***
14	500	PDBN-3	(2,3,9)	=	5.41	**0.1***
14	2000	PDBN-3	(2,3,9)	=	4.96	**0.04***
14	5000	PDBN-3	(2,3,9)	=	4.76	**0.02***
18	100	DBN	(1,8,9); (9)	$\not\subseteq$	2.17	2.17
18	500	PDBN-3	(1,8,9)	=	1.85	**0.1***
18	2000	PDBN-3	(1,8,9)	=	1.7	**0.03***
18	5000	PDBN-3	(1,8,9)	=	1.52	**0.02***
Time series length = 30						
2	100	PDBN-3	(15,17,29)	=	1.97	**0.1***
2	500	PDBN-3	(15,17,29)	=	1.93	**0.02***
2	2000	PDBN-6	(15,17,29); (1,6,15,17,22,29)	$\subseteq$ +3	1.92	**0.02***
2	5000	PDBN-5	(15,17,29); (3,15,16,17,29)	$\subseteq$ +2	1.92	**0.01***
6	100	PDBN-3	(18,19,29); (17,19,29)	$\not\subseteq$	17.15	**1.53***
6	500	PDBN-3	(18,19,29)	=	17.09	**0.05***
6	2000	PDBN-3	(18,19,29)	=	17.13	**0.03***
6	5000	PDBN-3	(18,19,29)	=	17.12	**0.02***
10	100	PDBN-3	(20,24,29)	=	25.57	**0.38***
10	500	PDBN-3	(20,24,29)	=	25.69	**0.07***
10	2000	PDBN-3	(20,24,29)	=	25.59	**0.05***
10	5000	PDBN-3	(20,24,29)	=	25.09	**0.03***
14	100	PDBN-3	(8,21,29)	=	15.53	**0.47***
14	500	PDBN-3	(8,21,29)	=	15.3	**0.17***
14	2000	PDBN-3	(8,21,29)	=	15.15	**0.07***
14	5000	PDBN-3	(8,21,29)	=	15.05	**0.04***
18	100	PDBN-3	(1,17,29)	=	12.63	**0.61***
18	500	PDBN-3	(1,17,29)	=	12.07	**0.11***
18	2000	PDBN-3	(1,17,29)	=	12.03	**0.06***
18	5000	PDBN-3	(1,17,29)	=	11.97	**0.05***

Table 6.3: Simulations with *data generated from PDBN-3 models*. The best KL divergence values are given in bold face and followed by an asterisk.

datasets (i.e. those with $d = 100$ sequences) were simpler than the reference models used to generate simulated data in virtually every case. Hence, the heuristic tends to operate in a conservative mode when there is scarcity of data. This also indicates that the methodology was effective in combating overfitting in these simulations.

With regard to the structural partitioning and quality measurements for these models: (1) the cuts of the learned models were all part of the cut sets of the reference models in almost all cases (note that this includes all the cases with a $\not\subseteq$); and (2) the divergences of the learned PDBNs were substantially smaller than those of DBNs, specially when data was generated from PDBN-3 models,

n	KL(DBN) - KL(L)	KL(L)	'$=$' (total)	'$\subseteq +a$'	'$\not\subseteq$' (total)
Time series length = 10					
2	0.95	0.04	5(12)	1.5	1(12)
6	1.58	0.06	12(12)	0	0(12)
10	1.02	0.29	10(12)	0	2(12)
14	2.49	0.23	11(12)	0	1(12)
18	0.63	0.36	10(12)	0	2(12)
Time series length = 30					
2	2.31	0.03	7(12)	2.6	0(12)
6	10.82	0.17	11(12)	0	1(12)
10	10.81	0.1	12(12)	0	0(12)
14	8.02	0.14	12(12)	0	0(12)
18	8.2	0.15	12(12)	0	0(12)

Table 6.4: Summary of simulations with DBNs and PDBNs. Abbreviations: **L** = learned model (heuristic), **KL (M)** = KL divergence between model M and the reference model. Positive values in the 2nd column indicate higher divergences achieved by DBNs. The 4th, 5th and 6th columns refer to the structural comparison of Section 6.4.1 and stand for the number of equal cut sets, average number of additional cut sets in learned models, and number of remaining cases respectively.

indicating a decent learning ability of the heuristic in the difficult situation of small datasets.

6.5 LEARNING TEMPORAL MODELS OF PSYCHOTIC DEPRESSION

6.5.1 *Bayesian networks in psychiatry*

The use of probabilistic graphical models in psychiatry has been fairly narrow. Existing research is mainly restricted to semi-automatic and fully handcrafted approaches, namely, learning only the parameters from data [41, 157] and eliciting both structure and parameters from descriptive statistics and expert knowledge [49, 103]. Although making use of expert knowledge might be necessary, e.g. in order to include established medical knowledge, the use of a data-driven approach has been able to discover new and unexpected insights in a multitude of fields. Furthermore, an advantage of BN models that can be of interest in psychiatry studies lies on making predictions when provided with incomplete evidence (e.g. only a few symptoms). This feature has been explored in some studies [49, 103], however at the individual level of a few patients (whether real or artificial), consequently, there is still a need for understanding associations between different variables in a more comprehensive and systematic way. This can include inferences for a population of patients, in order to reveal more general knowledge about, for example, the predictive power among different sets of features.

In the literature on BNs in psychiatry, so far time has not been a factor that has been taken into account in a comprehensive manner. Except for [41] that deals with the beginning and end of treatment, research that considers a broad range of time granularities has not been done up to this moment. This could be of interest, e.g., to controlled treatment trials and longitudinal diagnosis, where the examination of some form of history or time series measurements would allow for a more global comprehension of, for example, the evolution of mental illnesses and a more accurate diagnosis. For prediction with BNs and extensions such as DBNs, it is not required to enter all the symptoms as input for these models to be able to deliver predictions about the future. Furthermore, these predictions can be done for any point in future. Besides prediction, temporal models can also be used to find associations taking into account the time dimension. On the other hand, well-known models such as regression seem to be less flexible with regard to tasks such as the mentioned ones.

Within the field of psychiatry, diseases that have been covered under a BN approach include depression [36, 41, 103], social anxiety [157], schizophrenia [49], as well as analyzing the use of BNs on diagnosis in psychiatry [162]. Moreover, there is little research on using temporal models for better understanding psychotic depression, which besides being a severe mental disorder, brings an additional complexity due to the presence of psychosis and depression factors.

6.5.2 *Problem description and data*

To illustrate the use of non-homogeneous probabilistic models and the heuristic construction procedure proposed in this work, a case study in psychiatry is considered. It comprises a dataset from an original study designed to assess three different drugs to treat psychotic depression over 7 weeks [175]. The primary outcome of the original study aimed at comparing the drugs to depression levels and psychotic features at treatment endpoint. In this work, we aimed at answering a different research question: *to which extent do depressive and psychotic symptoms interact over time*? To this end, temporal models as DBNs and PDBNs are used to evaluate a large range of hypothesis about PD while modeling explicit relationships between psychotic and depressive features. We first discuss the results obtained by the heuristic algorithm when applied over psychiatry data, aiming at: (1) a more technical perspective based on fitting assessment between DBNs and PDBNs; and (2) an investigation of the dependences in the graphical structure. Then, in Section 6.6 we make use of the obtained models to answer clinically-oriented research questions, as the one mentioned earlier.

Differently from the original study, in which the primary outcome was the sum of the 17-item Hamilton depression rating scale (abbreviated as HDRS17) [81], in this section we considered the individual symptoms of the HDRS17. The dataset consists of 122 patients' data, from which 100 are patients that completed the treatment. Given the limited data, we used the 6-item melancholia sub-scale (HDRS6) [89] instead of the complete HDRS17, consisting of the features shown on Table 6.5. Using the melancholia sub-scale is, therefore, two-fold: it avoids

the usage of the complete HDRS17 upon the available scarce dataset, whereas HDRS6 is able to capture the core symptoms of depression [89]. In addition, two psychotic features were considered (hallucinations and delusions), totalizing eight features.

Psychiatry dataset [175]	
Number of sequences (complete)	122 (100) patients
Number of time points	8 (including baseline)
Depression features (HDRS6)	Depressed mood (Dm), Guilt (Gu), Work and Activities (Ac), Psychomotor Retardation (Re), Psychic Anxiety (Ap), and Somatic General (Sg)
Psychotic features	Hallucinations (Ha) and Delusions (De)
Study's period and location	2002-2007, The Netherlands

Table 6.5: Summary of psychiatry data.

The *somatic general* item takes values from the set $\{0, 1, 2\}$, where the value 0 means the item is *absent*, and the value 2 means it is *clearly present*. The other items of HDRS6 are graded on $\{0, 1, 2, 3, 4\}$, where 0 means the item is *absent*, and 4 means the item is *severe* [81]. To use as much data as possible, the incomplete cases were imputed with the same method used in the original study [175], namely, the last observation carried forward (LOCF). The frequencies of the imputed data at each week are shown on Table 6.6. An additional step in data preprocessing to cope with the limitation of dataset size consisted of discretizing each item as binary variables on $\{low, high\}$, as follows: $\{0, 1\}$ was mapped to *low*, while $\{2, 3, 4\}$ (for five-valued variables) and $\{2\}$ (for the three-valued variable) were mapped to *high*.

6.5.3 *Heuristic learning*

Applying the heuristic procedure over the data first yields a DBN, with mean log-likelihoods -351.18. In the first iteration of the heuristic refinement, it tries to find a model with two cuts that is a better fit than the DBN, which in fact was possible, precisely a PDBN-2 with cuts $\{4, 7\}$ and fit of -345.53, as show on left side of Fig. 6.2. Although not expanded further, the model with cuts $\{6, 7\}$ was also a better fit than the DBN (mean equal to -350.31). Since the algorithm found an improvement over the current best solution (the DBN), it updates the best solution to the most fit PDBN-2 and continues the heuristic search, now over PDBNs-3. As the right plot of Fig. 6.2 shows, the search again could find an improved solution, precisely a PDBN-3 with an additional cut just before the last cut, leading to a new cut set $\{4, 6, 7\}$ and mean log-likelihood of -344.80. Consequently, a new iteration is began over PDBNs-4, however, no further improvement could be achieved this time since the best fitting PDBN-

	Depressed mood						Guilt					
t	**0**	**1**	**2**	**3**	**4**	μ	**0**	**1**	**2**	**3**	**4**	μ
0	0	0	0.04	0.35	0.61	**3.57**	0.04	0.05	0.14	0.14	0.63	**3.27**
1	0.01	0.02	0.14	0.41	0.43	**3.23**	0.04	0.07	0.2	0.23	0.45	**2.98**
2	0.05	0.07	0.26	0.39	0.23	**2.69**	0.09	0.15	0.24	0.2	0.33	**2.52**
3	0.1	0.13	0.26	0.29	0.22	**2.4**	0.15	0.23	0.25	0.16	0.22	**2.07**
4	0.16	0.17	0.3	0.2	0.17	**2.07**	0.24	0.23	0.2	0.14	0.19	**1.81**
5	0.22	0.16	0.23	0.22	0.17	**1.97**	0.3	0.2	0.2	0.12	0.17	**1.67**
6	0.25	0.12	0.27	0.2	0.15	**1.87**	0.34	0.16	0.18	0.15	0.17	**1.66**
7	0.26	0.15	0.26	0.2	0.13	**1.79**	0.34	0.23	0.16	0.1	0.17	**1.52**

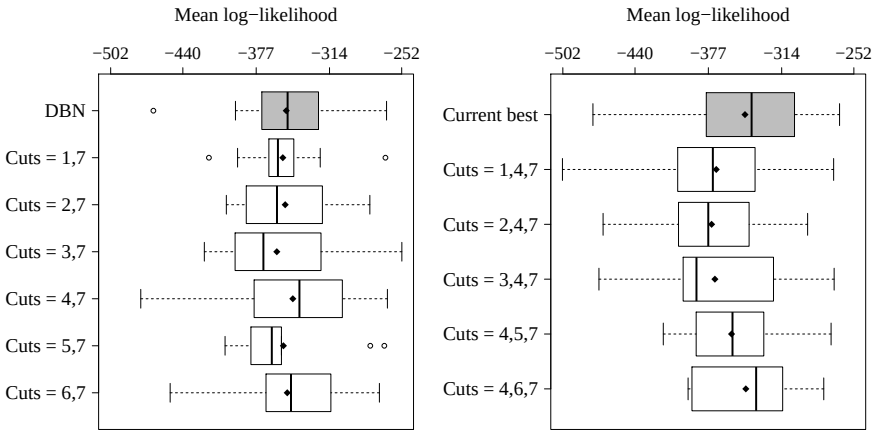
	Psychomotor retardation						Psychic anxiety					
t	**0**	**1**	**2**	**3**	**4**	μ	**0**	**1**	**2**	**3**	**4**	μ
0	0.16	0.3	0.31	0.22	0.02	**1.65**	0.03	0.14	0.27	0.37	0.19	**2.54**
1	0.15	0.33	0.34	0.16	0.02	**1.59**	0.11	0.16	0.29	0.29	0.16	**2.22**
2	0.27	0.3	0.29	0.12	0.02	**1.34**	0.18	0.22	0.3	0.23	0.07	**1.8**
3	0.33	0.35	0.22	0.08	0.02	**1.11**	0.29	0.25	0.23	0.16	0.07	**1.47**
4	0.4	0.31	0.2	0.07	0.02	**0.98**	0.3	0.26	0.2	0.17	0.06	**1.42**
5	0.53	0.21	0.18	0.06	0.02	**0.81**	0.39	0.2	0.24	0.12	0.05	**1.24**
6	0.52	0.27	0.13	0.06	0.02	**0.77**	0.39	0.16	0.23	0.17	0.04	**1.3**
7	0.62	0.18	0.12	0.06	0.02	**0.66**	0.38	0.26	0.19	0.12	0.05	**1.2**

	Work and activities						Somatic general			
t	**0**	**1**	**2**	**3**	**4**	μ	**0**	**1**	**2**	μ
0	0	0	0.15	0.49	0.36	**3.21**	0.1	0.3	0.61	**2.54**
1	0	0	0.21	0.52	0.27	**3.06**	0.16	0.34	0.51	**2.22**
2	0	0.02	0.34	0.5	0.14	**2.76**	0.22	0.43	0.34	**1.8**
3	0.01	0.08	0.35	0.4	0.16	**2.61**	0.34	0.39	0.27	**1.47**
4	0.02	0.12	0.4	0.34	0.12	**2.43**	0.27	0.48	0.25	**1.42**
5	0.02	0.14	0.43	0.29	0.12	**2.34**	0.39	0.42	0.2	**1.24**
6	0.03	0.19	0.36	0.3	0.12	**2.29**	0.41	0.38	0.21	**1.3**
7	0.07	0.25	0.37	0.2	0.11	**2.03**	0.4	0.36	0.24	**1.2**

Table 6.6: Relative frequencies of HDRS6 items of psychiatry data at each week, where μ denotes the respective weighted means.

4 had a mean of -362.61 (plot not shown), leading to the termination of the procedure. Hence, the model returned was a PDBN-3 with cuts at $\{4, 6, 7\}$.

A more detailed examination of the time partitioning of the resulting PDBN-3 can reveal insight on the underlying dynamics of the psychiatric treatment. In general lines, it suggests that the dynamics governing roughly the first half of the treatment's duration is distinguished from the remaining weeks. The second half of treatment is further dichotomized since the transition pattern to the last week is distinguished as well. Hypothesis can be devised from this structural partitioning, e.g. whether there are one or more symptoms that have stronger influence on the others in the first stage, and whether the last transition is distinguished due to a possible stabilization. Nonetheless, clinically relevant questions as these need a stronger assessment based on the graphical structure and distributions of each of the three components of the model, as covered in the next section.



(a) First iteration: 2 cuts.

(b) Second iteration: 3 cuts.

Figure 6.2: Boxplots for each stage of the heuristic over psychiatry data. The means are represented by a diamond symbol.

6.5.4 *Transition structures*

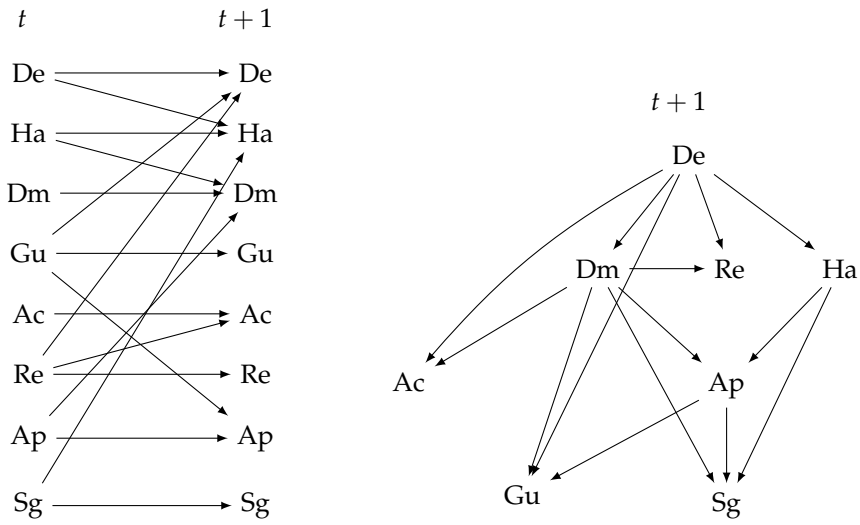
The structure of the DBN is shown in Fig. 6.3, while the structure of the conditional BNs that compose the PDBN-3 are shown in Figures 6.4 and 6.5. For a clearer exposition, each conditional BN was split into *inter*-temporal arcs (i.e. those from $t+1$ to t) and *intra*-temporal arcs (those delimited to each point $t+1$). Note that DBN's and PDBN-3's initial structure are naturally the same. Both models indicate the existence of a self-influence for every feature when moving from present to future. More precisely, if A is a feature, the chain $A^{(t)} \rightarrow A^{(t+1)}$ has been regularly learned for both DBN and PDBN-3, indicating (part of) the direct effect received by $A^{(t+1)}$.

6.6 MODEL ASSESSMENT FROM A CLINICAL PERSPECTIVE

In this section we approach the use of the learned models for psychotic depression, specially the DBN and the PDBN-3, to support answering clinically-oriented questions.

6.6.1 *Marginals of symptoms over time*

The previous sections showed that the PDBN-3 learned by the heuristic procedure provided: a better fit and a richer transition structure information with respect to other evaluated PDBNs, including the DBN. A complementary and practical assessment of these models compare the marginal frequencies of each symptom per week, as seen in data, with the respective model-based marginal distributions. Table 6.7 presents the empirical and model-based marginals for each symptom



(a) Inter-temporal arcs. (b) Intra-temporal arcs.

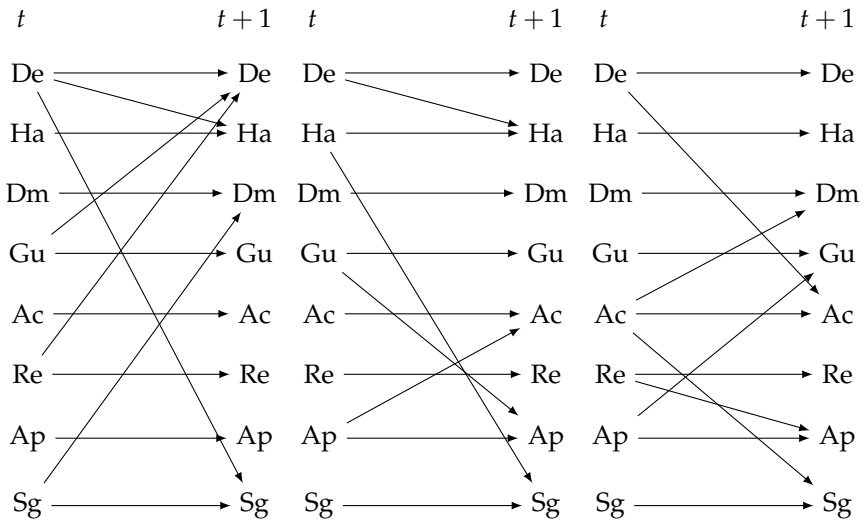
Figure 6.3: Structure of the DBN learned from the psychiatry data. Nodes on the left side of the inter-temporal arcs occur at time t, while those on the right at $t+1$. De = Delusions, Ha = Hallucinations, Dm = Depressed mood, Gu = Guilt, Ac = Work and activities, Re = Psychomotor retardation, Ap = Psychic anxiety, Sg = Somatic general.

per week, where the value assumed is either *true* or *high*. A summary of this information is presented at Table 6.8.

Concerning the psychotic symptoms, the PDBN-3 produced marginals that are closer to the empirical data than the DBN on average. With respect to depressive symptoms, a superior fit was achieved by the PDBN-3, except for the symptom psychomotor retardation.

6.6.2 *Predictive symptoms over time*

As discussed before, selecting an adequate structure is an important step to capture the underlying distribution in data as precisely as possible. As a probabilistic graphical model, the structure of PDBNs can be systematically verified for statistical independences among two sets of random variables by means of d-separation properties [104], essentially testing the paths between the respective nodes in the structure. As the Figures 6.4 and 6.5 show, the marginal statistical dependences, both direct and indirect (i.e. through paths with two or more arcs), dominated over the marginal independences. Nevertheless, the independence relation $\perp\!\!\!\perp_P$ (or its counterpart $\not\perp\!\!\!\perp_P$) is qualitative, in the sense that two variables being dependent does not directly inform about any intensity in which this dependence occur.

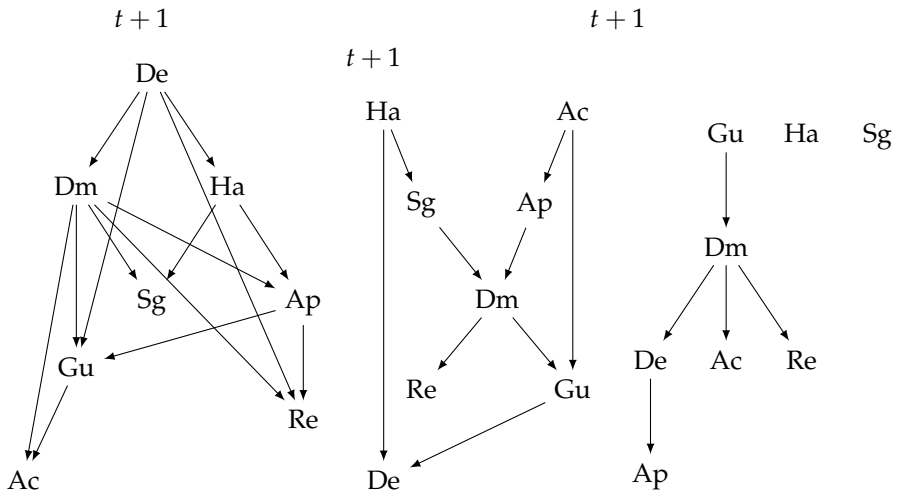


(a) First cut: $[0, 4]$. (b) Second cut: $[4, 6]$. (c) Third cut: $[6, 7]$.

Figure 6.4: *Inter*-temporal arcs of the PDBN-3 learned from the psychiatry data. De = Delusions, Ha = Hallucinations, Dm = Depressed mood, Gu = Guilt, Ac = Work and activities, Re = Psychomotor retardation, Ap = Psychic anxiety, Sg = Somatic general.

In this context, we approach a research question within the field of psychiatry, specially in psychotic depression: *to which extent do psychotic and depressive features interact during treatment*? This question can be rephrased more concretely as: *how predictive are the psychotic symptoms to depressive symptoms, and vice-versa*? To answer this question, statistical (in)dependences play a key role, since it is the fundamental criterion to decide on dependence and independence. However, it must be complemented to allow an assessment of the intensity of dependence among different dependent variables, aiming ultimately at discovering adequate predictors, i.e. features capable of performing an effective prediction of the interested symptoms. Intuitively, a symptom is a good predictor if each of its groups (i.e. its values) induces a different distribution on the predicted symptom; in other words, it should allow to reasonably distinguish the predicted symptom.

In this section, the odds ratio criterion is employed to determine the strength of predictors. A subset of time points was selected as conditioning points to observe a psychotic (resp. depressive) symptom and then compute the ORs of future time points for each depressive (resp. psychotic) symptom. Using multiple points allows to evaluate the dynamics of predictive capability as treatment progresses and more information become available. These conditioning points were selected to match approximately the cut points of the PDBN-3 learned heuristically, namely, $\{1, 4, 6\}$. The baseline point ($t = 0$) was discarded since it was a weak predictor for most of these predictions.



(a) First cut: $[0, 4]$. (b) Second cut: $[4, 6]$. (c) Third cut: $[6, 7]$.

Figure 6.5: *Intra*-temporal arcs of the PDBN-3 learned from the psychiatry data. De = Delusions, Ha = Hallucinations, Dm = Depressed mood, Gu = Guilt, Ac = Work and activities, Re = Psychomotor retardation, Ap = Psychic anxiety, Sg = Somatic general.

In order to compute an OR, suppose X is a psychotic symptom observed at some point (e.g. at $t = 1$), and Y is a depressive symptom that will be predicted at $t = i$, $i > 1$; therefore, $\text{dom}(X) = \{true, false\}$ and $\text{dom}(Y) = \{low, high\}$. Then, the odds ratio to predict Y given X is:

$$\text{OR}(Y^{(i)}|X^{(1)}) = \frac{\text{odds}(Y^{(i)} = high \mid X^{(1)} = true)}{\text{odds}(Y^{(i)} = high \mid X^{(1)} = false)} \tag{6.4}$$

$$= \frac{\dfrac{P(Y^{(i)} = high \mid X^{(1)} = true)}{1 - P(Y^{(i)} = high \mid X^{(1)} = true)}}{\dfrac{P(Y^{(i)} = high \mid X^{(1)} = false)}{1 - P(Y^{(i)} = high \mid X^{(1)} = false)}} \tag{6.5}$$

We fix that each depressive variable Y is predicted with level *high*, hence, the OR indicates the chances of having level *high* in the future according to each group of a psychotic feature X. If OR > 1, then it is more likely that the depressive feature Y will have level *high* if the patient comes from the group with $X = true$ compared to the patients coming from the group $X = false$; if OR < 1, it is more likely to observe Y at *high* in the group $X = false$ than in the group $X = true$; finally, if OR $= 1$, there is no association between X and Y, i.e. knowing the group of this particular psychotic feature does not affect the predictions for this depressive symptom. For the sake of terminology, an OR > 1 is also called a positive correlation, while an OR < 1 indicates a negative correlation. Note that

Symptom	Marginal probability (%)							
	t=0	t=1	t=2	t=3	t=4	t=5	t=6	t=7
Delusions								
Data	91.0	72.1	59.0	47.5	40.2	36.1	32.0	30.3
DBN	-0.09	**0.43**	**0.32**	2.03	1.93	**0.22**	**-0.18**	-1.95
PDBN-3	-0.09	-0.88	-1.32	**0.28**	**0.16**	0.38	1.6	**0.11**
Hallucinations								
Data	23.8	15.6	16.4	13.1	13.1	11.5	13.9	11.5
DBN	0.03	3.69	**-0.25**	0.68	**-1.06**	-0.77	-4.26	-2.62
PDBN-3	0.03	**2.77**	-1.59	**-0.58**	-1.95	**-0.01**	**-2.05**	**-1.66**
Depressed mood								
Data	100.0	97.5	88.5	77.0	67.2	62.3	62.3	59.0
DBN	-0.83	**-4**	**-2.22**	2.02	4.96	4.07	**-1.07**	-2.06
PDBN-3	-0.83	-4.39	-3.66	**-1.08**	**0.67**	**4.02**	2.5	**1.76**
Guilt								
Data	91.0	88.5	76.2	62.3	53.3	50.0	50.0	42.6
DBN	-0.03	**-5.78**	**-2.37**	3.09	4.56	**1.49**	-3.76	-0.84
PDBN-3	-0.03	-6.72	-3.92	**0.9**	**2.03**	3	**1.17**	**-0.07**
Activities								
Data	100.0	100.0	98.4	91.0	86.1	83.6	77.9	68.0
DBN	-0.83	-4.36	-6.87	-3.87	-3.13	-4.52	**-2.36**	4.47
PDBN-3	-0.83	**-3.03**	**-4.14**	**0.16**	**1.73**	**-0.18**	2.72	**2.66**
Retardation								
Data	54.9	52.5	43.4	32.0	28.7	25.4	20.5	19.7
DBN	-0.1	-6.18	-4.38	**1.32**	**-0.01**	**-0.41**	1.77	**0.39**
PDBN-3	-0.1	**-4.3**	**-2.96**	1.78	-0.45	-2.73	**-0.86**	-2.32
Psychic anxiety								
Data	82.8	73.0	59.8	45.9	43.4	41.0	44.3	36.1
DBN	-0.01	**-4.76**	**-1.04**	5.93	3.04	**1.07**	-5.76	**-0.57**
PDBN-3	-0.01	-5.54	-3.19	**2.87**	**-0.56**	3.36	**0.17**	1.98
Somatic general								
Data	60.7	50.8	34.4	27.0	25.4	19.7	21.3	23.8
DBN	-0.02	**-6.71**	**0.83**	3.03	1.2	4.47	**0.94**	**-3.05**
PDBN-3	-0.02	-7.28	-0.95	**1.15**	**-0.29**	**1.57**	-1.63	-3.33

Table 6.7: Marginal distributions over time: psychiatry data and learned models (the latter minus the former). The time span is split according to the cut set of the PDBN-3.

for the case when X is depressive and Y is psychotic, we fix *true* for X, and *high* and *low* in the numerator and denominator for Y respectively.

Additionally, to evaluate of the significance of the association between each X and Y, tables of contingency were constructed based on expected counts from the model. The Fisher's exact test was employed to evaluate the statistical significance of these, under a significance level of $\alpha = 0.05$.

6.6.2.1 *Predictors for depression*

Table 6.9 shows the ORs for psychotic features one week after baseline (i.e. at $t = 1$), acting as predictors for depression. These results suggest that delusions at that point had an at least reasonable association with the symptoms depressed mood and guilt, i.e. for at least half of the future points that were predicted. On the other hand, hallucinations at $t = 1$ showed to be less associated to the

Symptom	Mean Diff. (DBN)	Mean Diff. (PDBN-3)
Delusions	0.89	**0.6**
Hallucinations	1.67	**1.33***
Depressed mood	2.65	**2.36***
Guilt	2.74	**2.23***
Activities	3.8	**1.93***
Retardation	**1.82***	1.94
Psychic anxiety	2.77	**2.21***
Somatic general	2.53	**2.03***

Table 6.8: Summary of percentage differences of learned models to the marginal frequencies of psychiatry data. The absolute values are used to compute the means.

depressive symptoms. Nonetheless, somatic general contrasts with this pattern, as it has been predicted by hallucinations almost until the end of the remaining weeks of treatment. The other case where some dependency on this predictor was noticed is psychic anxiety, however for a shorter period of time (three weeks forward).

With respect to the predictive power of psychotic symptoms observed at $t = 4$ and $t = 6$ (Table 6.10, left and right respectively), delusions stood as predictor of depressed mood and guilt, in this situation as a stronger predictor (all three future predictions were significant). Other depressive symptoms were mostly weakly associated to delusions. Hallucinations at these time points showed a more restricted behavior than before, since it acted as predictor of somatic general only, although by significant associations.

Symptom & predictor	t=2	t=3	t=4	t=5	t=6	t=7
Depressed mood						
Delusions$^{(1)}$	**5.15***	**3.39***	**2.72***	1.75	1.38	1.44
Hallucinations$^{(1)}$	1.13	1.5	1.46	1.59	1.66	1.48
Guilt						
Delusions$^{(1)}$	**3.84***	**3.27***	**2.75***	**2.11***	1.84	1.62
Hallucinations$^{(1)}$	1.1	1.12	1.2	1.2	1.3	1.29
Activities						
Delusions$^{(1)}$	3.53	2.23	2.45	1.42	1.4	1.45
Hallucinations$^{(1)}$	1.34	1.04	1.38	1.38	1.6	1.47
Retardation						
Delusions$^{(1)}$	**3.24***	**3.22***	2.4	2.02	1.67	1.35
Hallucinations$^{(1)}$	1.15	1.16	1.24	1.33	1.25	1.35
Psychic anxiety						
Delusions$^{(1)}$	1.33	1.21	1.16	1.27	1.33	1.46
Hallucinations$^{(1)}$	**2.54***	**2.66***	**2.41***	1.65	1.32	1.31
Somatic general						
Delusions$^{(1)}$	0.96	0.95	0.8	0.7	0.64	0.82
Hallucinations$^{(1)}$	**3.31***	**3.27***	**2.86***	**3.07***	**2.97***	2.23

Table 6.9: Odds ratios for **psychotic symptoms as predictors**. An OR greater than 1 indicates that the level *high* on the depressive feature is more likely to be observed in the group *true* than in the group *false* of the psychotic feature. Results marked in bold and * stand for a statistically significant association.

Symptom & predictor	t=5	t=6	t=7
Depressed mood			
Delusions$^{(4)}$	**3.09***	**2.26***	**2.17***
Hallucinations$^{(4)}$	1.98	2.14	1.71
Guilt			
Delusions$^{(4)}$	**4.15***	**2.93***	**2.34***
Hallucinations$^{(4)}$	1.19	1.31	1.4
Activities			
Delusions$^{(4)}$	2.26	1.81	**2.59***
Hallucinations$^{(4)}$	2.52	1.53	1.61
Retardation			
Delusions$^{(4)}$	**2.97***	2.02	1.98
Hallucinations$^{(4)}$	1.4	1.25	1.36
Psychic anxiety			
Delusions$^{(4)}$	1.88	1.88	**2.21***
Hallucinations$^{(4)}$	2.18	1.53	1.45
Somatic general			
Delusions$^{(4)}$	0.97	0.87	0.99
Hallucinations$^{(4)}$	**6.52***	**6.18***	**4.91***

Symptom & predictor	t=7
Depressed mood	
Delusions$^{(6)}$	**2.72***
Hallucinations$^{(6)}$	1.67
Guilt	
Delusions$^{(6)}$	**3.62***
Hallucinations$^{(6)}$	1.2
Activities	
Delusions$^{(6)}$	**5.66***
Hallucinations$^{(6)}$	1.61
Retardation	
Delusions$^{(6)}$	2.04
Hallucinations$^{(6)}$	1.34
Psychic anxiety	
Delusions$^{(6)}$	**3.52***
Hallucinations$^{(6)}$	1.25
Somatic general	
Delusions$^{(6)}$	1.14
Hallucinations$^{(6)}$	**4.31***

Table 6.10: Odds ratios for **psychotic symptoms as predictors** (cont.). Left: $t = 4$, right: $t = 6$.

6.6.2.2 *Predictors for psychosis*

In the following, we evaluate how predictive the depressive symptoms are to predict psychotic symptoms. Note that ORs are not symmetric; for example, we calculate $P(\text{Som.gen}^{(t)}|\text{Del}^{(0)})$ to assess whether delusions is predictive to somatic general, while we compute $P(\text{Del}^{(t)}|\text{Som. gen}^{(0)})$ to assess whether somatic general is predictive to delusions. Note that these two might represent distinct quantities.

Table 6.11a shows the odds ratio for each depressive symptom observed at $t = 1$. As the results indicate, the depressive symptoms were not significantly strong to predict delusions, except depressed mood, guilt and retardation, which accounted for a weak association (precisely, two weeks ahead of the reference measurement). Regarding hallucinations, there is virtually no depressive symptom predictor for the case of $t = 1$.

On the other hand, updating the depressive symptoms at $t = 4$, as shown on Table 6.11b (left), increased the association of the three symptoms mentioned before to predict delusions until the end. The same insight applies to predict delusions at $t = 6$. Concerning the prediction of hallucinations, somatic general emerged with strong associations when measured both at $t = 4$ and $t = 6$, while psychic anxiety showed reasonable associations only when measured at the middle point, though.

6.7 CONCLUSIONS

In this work, we proposed a heuristic algorithm to learn non-homogeneous time dynamic Bayesian networks for relatively small temporal datasets with a small

Symptom & predictor	t=2	t=3	t=4	t=5	t=6	t=7
Delusions						
Depressed mood$^{(1)}$	**5.3***	**6.91***	5.04	4.2	3.66	3.21
Guilt$^{(1)}$	**2.91***	**2.86***	2.21	2.16	1.9	1.62
Activities$^{(1)}$	2.79	2.78	2.05	1.75	1.53	1.31
Retardation$^{(1)}$	**2.49***	**2.11***	1.8	1.57	1.37	1.38
Psychic anxiety$^{(1)}$	1.18	1.19	1.18	1.26	1.27	1.22
Somatic general$^{(1)}$	0.91	0.97	0.96	1.07	1.07	1.16
Hallucinations						
Depressed mood$^{(1)}$	0.58	0.49	0.42	0.83	0.9	0.75
Guilt$^{(1)}$	0.84	0.86	0.78	0.78	0.79	0.67
Activities$^{(1)}$	0.51	0.44	0.38	0.38	0.41	0.31
Retardation$^{(1)}$	1.08	0.93	0.91	0.81	0.93	1.07
Psychic anxiety$^{(1)}$	1.84	2.05	1.71	1.83	1.39	1.52
Somatic general$^{(1)}$	**3.04***	2.9	2.54	1.86	1.86	1.94

(a) Odds ratios based on $t = 1$.

Symptom & predictor	t=5	t=6	t=7
Delusions			
Depressed mood$^{(4)}$	**4.96***	**3.97***	**4.22***
Guilt$^{(4)}$	**8.13***	**5.62***	**4.58***
Activities$^{(4)}$	3.84	3.36	3.14
Retardation$^{(4)}$	**3.32***	**2.5***	**2.2***
Psychic anxiety$^{(4)}$	1.8	1.69	1.9
Somatic general$^{(4)}$	1.35	1.35	1.37
Hallucinations			
Depressed mood$^{(4)}$	1.2	1.2	0.97
Guilt$^{(4)}$	1.07	0.91	0.96
Activities$^{(4)}$	0.82	0.9	0.67
Retardation$^{(4)}$	1.04	1.3	1.27
Psychic anxiety$^{(4)}$	**3.8***	**3***	2.97
Somatic general$^{(4)}$	**4.85***	**3.6***	**3.36***

Symptom & predictor	t=7
Delusions	
Depressed mood$^{(6)}$	**3.94***
Guilt$^{(6)}$	**5.63***
Activities$^{(6)}$	1.83
Retardation$^{(6)}$	1.87
Psychic anxiety$^{(6)}$	**2.52***
Somatic general$^{(6)}$	1.19
Hallucinations	
Depressed mood$^{(6)}$	1.71
Guilt$^{(6)}$	1.35
Activities$^{(6)}$	1.25
Retardation$^{(6)}$	1.41
Psychic anxiety$^{(6)}$	1.29
Somatic general$^{(6)}$	**7.47***

(b) Odds ratios based on $t = 4$ (left) and $t = 6$ (right).

Table 6.11: Odds ratios for **depressive symptoms as predictors**. An OR greater than 1 indicates that the level *true* on the psychotic feature is more likely to be observed in the group *high* than in the group *low* of the depressive feature. Results marked in bold and * stand for a statistically significant association.

number of variables as typically encountered in clinical settings. Extensive simulations and a case study in psychiatry (psychotic depression) demonstrated its capability to find adequate models under different assumptions, which included data generated from non-homogeneous and homogeneous models. In particular, simulated experiments played an important role to show that, in more general scenarios, models based on non-homogeneous time have substantial benefits over DBNs on several aspects (e.g. model fit and problem insight) when the underlying process switches between different regimes on time. In the case of small datasets, common in many clinical studies, it was shown that the heuristic algorithm behaves in a more conservative fashion, i.e. it tends to produce slightly simpler non-homogeneous models compared to the reference models, and yet providing a decent fit.

Aiming at learning non-homogeneous models in the usually unfavorable scenario of data scarcity, an evaluation criterion employed by the heuristic explicitly

avoids over-specialized models, at the same time providing more robust models. Moreover, the search strategy of the heuristic, based on incremental construction of non-homogeneous models, is able to cope with the trade-off between model complexity and data scarcity.

A first step towards a systematic application of probabilistic graphical models in psychiatry taking into account the temporal dimension was taken. It allowed to obtain insight about the dynamics of patient recovery in psychotic depression over the course of a controlled treatment. In particular, a research question aiming to answer the temporal relationship between psychotic and depressive features was investigated, supported by models learned with the heuristic procedure. The experimental assessment of the predictive capability of psychotic symptoms observed at different moments (near baseline, middle and near-end points) showed that the delusions symptom was more predictive than the hallucinations symptom on most cases. On the other hand, the depressive symptoms were less predictive for the psychotic symptoms. Nevertheless, a point to be observed is that in general the predictions were bidirectional, i.e. the symptoms from one category that stood as statistically significant predictors for the other can be interchanged.

Among future research, we intend to evaluate the developed algorithm in other real-world problems, as well as investigate further variations of the incremental search. For example, during the execution of the algorithm, different new solutions with equal or approximately equal score yet higher than the current best solution can be found; this is currently worked out by choosing one of these new solutions randomly and then resuming the search. The problem of handling multiple solutions is in fact recurring in the literature of Bayesian networks, where extensive research has been developed [33, 40, 108, 124]. In this direction, the approach of this chapter could benefit from such research, for example by extending the greedy search, as well as taking into account Bayesian approaches [145]. These further investigations could provide more insight about the distribution and the variance of the cut sets.

7

EXCEPTIONAL MODEL MINING USING DYNAMIC BAYESIAN NETWORKS

The discovery of subsets of data that are characterized by models that differ significantly from the entire dataset, is the goal of exceptional model mining. This task has clear relevance nowadays, facing the current need for interpretable AI models. In this chapter, we introduce temporal exceptional model mining *to capture not only multiple targets, but also complex temporal relationships among the targets. Temporal exceptional model mining opens new avenues for discovering groups that deviate from the crowd, in domains such as medical treatments and industrial processes, where repeated measurements of a set of variables might be available. The contributions of this chapter are three-fold: (i) a new definition of the task of temporal exceptional model mining is provided; (ii) we characterize the discovery of exceptional dynamic Bayesian networks by means of a new interestingness score, and (iii) the practical value of the proposed method is demonstrated based on process data of funding applications and by comparisons with previous EMM approaches.*

7.1 INTRODUCTION

Subgroup discovery (SD, for short) is the task of identifying subsets of a dataset that have unusual distributions with respect to a target variable [87]. Subgroup discovery and clustering have different goals [181] as clustering seeks subsets of data that are internally homogeneous, while in SD the models that allow for *interpreting differences* are sought, as they support explaining why an object belongs to a subgroup. Interpretability is essential in artificial intelligence, even with successful, yet less interpretable models as deep neural networks [88, 114], which justifies the relevance of SD research.

In many real-world applications one has to deal with multiple and complex targets. This has led to the generalization of subgroup discovery known as *exceptional model mining* (EMM, for short) [110]. EMM aims to identify subgroups with models fitted on the targets that are unusual compared to a reference model (typically the model fitted on the whole dataset).

The computational burden of SD lies in subgroup search [87], as determining whether a subgroup is unusual is often straightforward. In EMM, however,

models over multiple variables are fitted on subgroup data, which results in a two-fold challenge: (*i*) the choice of suitable model classes, as model learning is now an integral part of the framework, and (*ii*) how to determine whether a subgroup model is exceptional. This increased complexity has been compensated by the discovery of useful exceptional models, e.g., based on linear regression [110], Bayesian networks [57], subgraph mining [9, 100], social networks [4] and preferences [149].

Remarkably, little research has been done in exploiting temporal submodels for EMM. Submodels based on Markov chains (MCs, for short) have been investigated [112], as well as latent variable-modeling by means of hidden Markov models [161]. In this chapter, we introduce the discovery of temporal submodels by means of the *temporal exceptional model mining* task (TEMM, for short), which is demonstrated by means of dynamic Bayesian networks (DBNs, for short) as model class. We argue that using DBNs allows for a general and intuitive representation of subgroups obtained from multiple and temporal observations. The DBN representation allows for extra, qualitative information that can be gleaned from the model structure.

The contributions of this chapter are as follows. First, the novel task of temporal exceptional model mining is defined, which can be seen as a generalization of previous research in EMM. Then we introduce the usage of DBNs for TEMM by proposing an *interestingness score* for identifying exceptional DBNs. Finally, the proposed methods are demonstrated by analyzing data of funding applications.

This chapter is organized as follows. In Section 7.2, we discuss the related work. In Section 7.3, we define the task of TEMM. In Section 7.4, we introduce a distance measure for exceptional DBNs. In Section 7.5 we present a search approach for exceptional DBNs. The experiments based on simulations and real data are discussed in Sections 7.6 and 7.7. The conclusions and future work are discussed in Section 7.8.

7.1.1 *Motivating example*

We describe next a running example which is also used in experiments with real-world data. In the European Union, farmers can apply for direct payments [56], which provide them additional income and incentives for sustainable production. A *funding application* is described by **Land Area**, **Young Farmer** (yes/no), and **Small Farmer** (yes/no). An application is submitted in a **Year** and is checked for eligibility by a **Department**. The work flow of an application is a sequence of *events* described by **Activity** and **Doc Type**.

We would like to know whether there are applications whose work flow (i.e. the dynamics of Activity and Doc Type) deviates considerably from the work flow of the whole population of applications. It could be the case that applications handled by a certain department take much longer than the average, or that applications submitted in a particular year have a specific work flow. By *automatically* discovering these subgroups, we could learn more about the process, which could e.g. help the organization to improve the process quality.

7.2 RELATED WORK

As a generalization of SD, exceptional model mining [58] is an active area of research and has been applied to different target variable representations. Earlier research includes the discovery of exceptional linear regression models [110] and the discovery of subgroups with Bayesian networks that have significant structural differences [57]. A more specialized application of EMM is tailored at sequential problems, yet over a single target, where discrete Markov chains with significantly different transition patterns have been investigated [112].

The aforementioned EMM research can be seen as *parameter-based approaches*, because subgroups are characterized based on the unusualness of some of the model parameters, e.g. regression slope, network structure, etc. On the other hand, model-based subgroup discovery [161] is an *evaluation-driven approach* that compares the distribution of subgroups by means of proper scoring rules.

Some body of research has dealt with *subgroup search*, whose aims include making the search more efficient, reducing the number of redundant subgroups, etc. Research has been done on providing bounds for some interestingness scores in the context of numerical targets that can be used for search pruning [111]. Subgroup search has also been formulated in terms of game theory [18], which allows for guiding the search toward the interestingness of subgroups while improving the lack of diversity that search might face.

Other extensions to SD and EMM operate on data other than the common attribute-value data. The approach in [113] is tailored for relational data and can extract very general structured patterns of subgroups. More recently, exceptional graph mining [9, 100] has been proposed to allow for the discovery of graph neighborhoods that are similar internally but exceptional to the general attributed graph (i.e. graphs with non-trivial vertices such as a list of attribute-value pairs) [9]. Research has been done on the discovery of exceptional social behavior from spatio-temporal [98], which helps understand networked interactions (e.g. as in how people interact in a neighborhood). Recently, EMM has been applied to finding subsets of data related to exceptional convolutional layers in convolutional neural networks [167], which might help the interpretation of such models.

7.3 TEMPORAL EXCEPTIONAL MODEL MINING

In this section we describe relevant background notions and define the task of temporal exceptional model mining.

7.3.1 *Temporal targets*

In order to represent subgroups in SD and EMM we define descriptor and target variables. The set of descriptor variables is a set $\mathbf{A}$ of random variables $\{A_1, \ldots, A_k\}$, where each A_i is a *descriptor variable* and has a domain $\mathrm{dom}(A_i)$. We denote values of the domain by lower-case letters such as $a_i \in \mathrm{dom}(A_i)$. In

standard SD, one normally models next to $\mathbf{A}$ a single variable X called *target variable*, while in EMM a *set of targets variables* $\mathbf{X} = \{X_1, \ldots, X_n\}$ is used instead. For example, in EMM for regression [110], the predictor and response variables are the target variables.

In TEMM, we assume that the target variables are the result of a temporal process that changes a set of basis variables $\mathbf{X}$.

Definition 7.1 (Temporal targets)**.** *Let $\mathbf{X}$ be a set of random variables. We assume that there is a process that changes $\mathbf{X}$ at regular time points, resulting in the variables $\mathbf{X}^{(0)}, \mathbf{X}^{(1)}, \ldots$. The variable $X_i^{(t)}$ denotes X_i at time t, and we denote by $X_i^{(t_1:t_2)}$ the variables X_i occurring from time t_1 up to t_2. The variables $X_i^{(t)}$, for $t \geq 0$, have the same domain. We call each $\mathbf{X}^{(t)}$ a temporal target.*

Based on Definition 7.1, we define the space of variables in TEMM as $\{\mathbf{A}, \mathbf{X}^{(0)}, \mathbf{X}^{(1)}, \ldots\}$. In practice, a data point in TEMM corresponds to configurations of $\mathbf{A}$ and a finite number of temporal targets. Based on this, we consider a multiset D of data points, where the ith data point is denoted by $(\mathbf{a}[i], \mathbf{x}[i]^{(0)}, \ldots, \mathbf{x}[i]^{(m_i)})$, in which m_i is the last temporal target. Thus, each data point of D has a particular number of temporal targets. An example is given next.

Example 7.1. *Consider the dataset for the application described in Section 7.1.1 with descriptors $\mathbf{A} = \{$**Year**, **Department**, **Number Parcels**, **Land Area**$\}$ and targets $\mathbf{X} = \{$**Activity**, **Doc Type**$\}$. Table 7.1 shows two data points of this dataset.*

7.3.2 *Subgroups*

A subgroup can be described by different pattern languages [57], depending on the type of data and the kind of patterns one wants to discover. In spite of different existing languages (see, e.g., [9, 113]), the attribute-value pattern language [58, 61, 128] is still very relevant in SD and EMM. In this work, we use this propositional language, which is defined based on the space of descriptor variables $\mathbf{A}$ as follows.

Definition 7.2 (Subgroup)**.** *Let $D = \{d_1, \ldots, d_m\}$ be a dataset (multiset) with each record d_i a collection of variable-value pairs $A_j[i] = a_j$ and $\mathbf{A} = \{A_1, \ldots, A_k\}$. Let φ denote an expression of the form $(A_{p_1} = a_{p_1} \wedge \cdots \wedge A_{p_q} = a_{p_q})$, where $\{p_1, \ldots, p_q\} \subseteq \{1, \ldots, k\}$. The subgroup associated to φ is defined as:*

$$G_\varphi = \left\{ d_i \in D \mid (A_{p_1}[i] = a_{p_1} \wedge \ldots \wedge A_{p_q}[i] = a_{p_q}) \right\} \tag{7.1}$$

We say that the number of descriptors of G_φ is q.

We refer to a subgroup defined by the expression φ either by G_φ or by the expression φ itself. For convenience, the domain of a binary descriptor such as A is denoted by $\mathrm{dom}(A) = \{a^-, a^+\}$. For example, an expression $(a_1^+ \wedge a_2^+ \wedge a_3^-)$ represents a subgroup with 3 binary descriptors. In Definition 7.2, a subgroup

Year	Department	# Parcels	Area	Activity	Doc Type
2016	4e	31	97.8	mail valid	Payment application
				initialize	Geo parcel document
				finish editing	Control summary
				performed	Reference alignment
				finish editing	Geo parcel document
				performed	Department control parcels
				finish editing	Geo parcel document
				calculate	Payment application
				decide	Payment application
				revoke decision	Payment application
				calculate	Payment application
				decide	Payment application
				begin payment	Payment application
				abort payment	Payment application
				begin payment	Payment application
				insert document	Payment application
				finish payment	Payment application

Year	Department	# Parcels	Area	Activity	Doc Type
2016	e7	37	97.8	mail valid	Payment application
				initialize	Geo parcel document
				finish editing	Control summary
				performed	Reference alignment
				performed	Department control parcels
				calculate	Payment application
				decide	Payment application
				revoke decision	Payment application
				calculate	Payment application
				decide	Payment application
				begin payment	Payment application
				insert document	Payment application
				finish payment	Payment application

Year	Department	# Parcels	Area	Activity	Doc Type
2017	6b	7	9.1	mail valid	Payment application
				pre-check	Geo parcel document
				finish editing	Control summary
				finish editing	Geo parcel document
				performed	Reference alignment
				initialize	Payment application
				finish editing	Geo parcel document
				calculate	Payment application
				finish editing	Geo parcel document
				calculate	Payment application
				decide	Payment application
				begin payment	Payment application
				insert document	Payment application
				finish payment	Payment application

Table 7.1: Data points of a process dataset, with $\mathbf{A} = \{\textbf{Year}, \textbf{Department}, \textbf{Number Parcels}, \textbf{Land Area}\}$ and $\mathbf{X} = \{\textbf{Activity}, \textbf{Doc Type}\}$. The temporal targets correspond to the work flow of events in the order they occurred.

is a subset of data points of D selected according to a propositional expression formed by a conjunction of attribute-value pairs. Not all attributes of $\mathbf{A}$ need to be involved in the subgroup expression, hence $p_q \leq k$. If $q = 1$ we say that the subgroup is *unitary*, otherwise the subgroup is *specialized*. The subscript φ is omitted from G_φ if no risk of ambiguity arises.

Each data point of G_φ is associated to a configuration of temporal targets for which notation is introduced next.

Definition 7.3 (Subgroup sequences)**.** *The subgroup sequences of a subgroup G_φ of D are given by:*

$$S(G_\varphi) = \{\mathbf{x}[i]^{(0:m_i)} \mid d_i \in G_\varphi\} \tag{7.2}$$

The size of subgroup G_φ is $\sum_{d_i \in G_\varphi} (m_i + 1)$ and is denoted by $|G_\varphi|$.

7.3.3 *Comparing subgroups*

In TEMM, a model shall be fitted on the subgroup's sequences. We refer to the model fitted on the data $S(G)$ of a subgroup G as its *subgroup model*. When we wish to compare subgroups in TEMM, we shall compare the subgroup models associated to these subgroups, hence this comparison is based on the space of temporal targets.

The notion of exceptional subgroups involves comparing subgroups based on some notion of distance. We define a distance notion with some *desirable properties* that serves as a basis for the development of distance measures for specific class of temporal models.

Definition 7.4 (Distance function)**.** *Given a multiset D, the distance function between two subgroups G and H of D is a real number denoted by $d(G, H)$. This distance has the following properties:*

$$d(G, H) \geq 0 \qquad \textit{non-negativity} \tag{7.3}$$

$$d(G, H) = 0 \textit{ if } G = H \qquad \textit{weak identity of indiscernibles} \tag{7.4}$$

$$d(G, H) = d(H, G) \qquad \textit{symmetry} \tag{7.5}$$

Other properties can be added to the above ones depending on the desired characteristics of the distance function. For example, by strengthening the second assumption and adding the triangle inequality, one would arrive at a distance function that would be a *metric*. The distance function should, however, be designed in such a way to support these properties.

7.3.4 *Exceptional subgroups*

One way to determine whether a subgroup G is exceptional is by considering a *reference* subgroup upon which the distance to G can be computed. We introduce

the notion of exceptional relation next, which has a few *desirable properties* of interest.

Definition 7.5 (Exceptional subgroup)**.** *Given a multiset D, we define a relation $ex \subseteq 2^D \times 2^D$, called exceptionality which has the following properties for two any subgroups G and H of D:*

$$ex(G,H) \implies ex(H,G) \quad \textit{(symmetry)} \tag{7.6}$$

$$\neg ex(G,G) \quad \textit{(anti-reflexive)} \tag{7.7}$$

If $ex(G,H)$ holds, we say that G is an exceptional subgroup with regard to the subgroup H.

The precise definition of which subgroups are exceptional depends on the definition of the distance function. An exceptionality relation will be defined in Section 7.4.4.

7.3.5 *Problem statement*

In TEMM, we wish to find all the subgroups G which are exceptional with regard to the population. One additional requirement is that every exceptional subgroup G must have a minimal size, i.e. $|G| \geq \sigma|D|$, where $\sigma \in [0,1]$ is the *minimal size threshold*. One can also specify some kind of preference for more specialized or more general subgroups (see, e.g., [112]).

7.4 EXCEPTIONAL DYNAMIC BAYESIAN NETWORKS

In this work, we consider dynamic Bayesian networks [68, 104, 124] as model class to represent subgroup models. We define a distance function for DBNs and instantiate it for a scoring function, allowing for the discovery of *exceptional dynamic Bayesian networks*.

7.4.1 *Dynamic Bayesian networks*

Dynamic Bayesian networks extend Bayesian networks for modeling processes with uncertainty. In this work, DBNs model the temporal targets from Definition 7.1.

In order to keep the model compact, a few assumptions are considered in dynamic systems such as DBNs. We say that a dynamic system over the temporal targets $\mathbf{X}$ is *Markovian* if $P(\mathbf{X}^{(t+1)} \mid \mathbf{X}^{(0:t)}) = P(\mathbf{X}^{(t+1)} \mid \mathbf{X}^{(t)})$, for all $t \geq 0$. This means that predicting the future state depends only on the current state. Another useful assumption is the *time homogeneity*, which holds in a dynamic system if the transitions $P(\mathbf{X}^{(t+1)} \mid \mathbf{X}^{(t)})$ are fixed for every $t \geq 0$. We refer the reader to Section 2.4 for more details on DBNs.

7.4.2 *Distance function*

Definition 7.6 (Mismatch score)**.** *Let D be a multiset over $\{\mathbf{A}, \mathbf{X}^{(0)}, \mathbf{X}^{(1)}, \ldots\}$ and G, H be two subgroups of D. Further, let us denote by M_G and M_H the dynamic Bayesian networks learned from G and H respectively by maximizing some scoring function. The mismatch score between M_G and M_H is:*

$$\begin{aligned} mismatch(M_G, M_H) = (score(M_G : G) - score(M_H : G)) \\ + (score(M_H : H) - score(M_G : H)) \end{aligned} \tag{7.8}$$

where score$(M : G)$ refers to the score of model M based on data G. The mismatch distance resembles the idea of learning and validation sets (e.g. as used in cross-validation [104])). However, here we are considering a more general situation, because we assume that G and H might not have come from the same distribution. In fact that is what we want to evaluate: the *error* that a model makes when given data not used to learn it. Intuitively, if the DBNs induced from G and H are similar one would expect a small mismatch value, while a high mismatch would be obtained had the models been too different. A few properties regarding the mismatch score are given next.

Proposition 7.1 (Weak identity of indiscernibles)**.** *Let M_G be a DBN fitted to the subgroup G of D. Then it holds that:*

$$mismatch(M_G, M_G) = 0 \tag{7.9}$$

Proof. Directly from the definition of mismatch score. □

Proposition 7.1 means that the weak identity of indiscernibles holds for the mismatch. However, it is not the case that a mismatch equal to zero implies that the subgroups G and H are the same. This is because D is a multiset, hence G and H might be associated to the same sequences while being two different subsets of D. Another relevant property is symmetry, which is formalised in the next proposition.

Proposition 7.2 (Symmetry)**.** *Given two DBNs M_G and M_H learned from two subgroups G and H of D, it holds that:*

$$mismatch(M_G, M_H) = mismatch(M_H, M_G) \tag{7.10}$$

Proof. Directly from the definition of mismatch score. □

A relevant property concerns the sign of the mismatch distance is given as follows.

Proposition 7.3 (Non-negativity)**.** *Let M_G and M_H be the DBNs learned from the subgroups G and H of D. Then it holds that:*

$$mismatch(M_G, M_H) \geq 0 \tag{7.11}$$

Proof. If $G = H$, the claim holds by Proposition 7.1. Otherwise, if M_G is the model learned from G, then it must hold that $\text{score}(M_G : G) \geq \text{score}(M_H : G)$ for any model M_H. This is because by Definition 7.6 M_G was learned by maximizing the score given the data G, then no other model can have better score given G. □

As the mismatch distance is non-negative, symmetric and has the weak identity of indiscernibles property, it follows that it can be taken as a distance function for TEMM, as discussed in Section 7.3.3.

7.4.3 *Scoring function*

In this work, we use Bayesian information criterion as scoring function (see Section 2.3.2), which is proportional to the log-likelihood of the model and includes a penalty to control for model complexity. For convenience, we repeat the definition of the BIC of a model M_G given data G as follows:

$$\text{BIC}(M_G : G) = 2 \log \mathcal{L}(M_G : G) - |M_G| \log |G| \tag{7.12}$$

where $\log \mathcal{L}(M_G : G)$ denotes the log-likelihood of the model M_G, $|M_G|$ the number of parameters of M_G, and $|G|$ is the number of observations of G. The negative value of the standard BIC was taken for the convenience of maximizing the score.

We assume that M_G is fitted by maximizing the BIC score as denoted by $\text{BIC}(M_G : G)$, and we shall denote by $\text{BIC}(M_G : H)$, with $H \neq G$, the score of M_G given data H different than that used to fit M_G. The BIC score corresponds to the score term of Definition 7.6.

7.4.4 *Exceptional subgroups*

We define next a general notion of exceptional DBNs.

Definition 7.7 (Exceptional subgroups)**.** *Consider the exceptionality relation $ex \subseteq 2^D \times 2^D$. We say that G is an exceptional subgroup with regard to a subgroup H, denoted by $ex(G, H)$, if the distribution of the DBN M_G is different from the distribution of the DBN M_H.*

Definition 7.7 implements the idea of exceptional subgroups delineated by Definition 7.5 applied to exceptional DBNs. It is straightforward to verify that the exceptionality relation just defined is symmetric and anti-reflexive, hence the relationship has the desired properties as discussed in Section 7.3.4.

In EMM, the reference subgroup used for determining the exceptionality of a subgroup is typically the full data D, also referred to as *population* [161]. This means that a subgroup of interest G would be compared with D, however, this comparison is made more convenient by instead comparing G with its complement denoted by $\bar{G}$ [57], which results in a comparison involving two disjoint subgroups. TEMM uses the population as reference subgroup as well,

thus for determining whether a subgroup G is exceptional we compare the subgroup models of G and $\bar{G}$.

7.5 IDENTIFYING EXCEPTIONAL SUBGROUPS

In this section, we discuss how the exceptionality of DBNs can be identified from data by considering reasonable assumptions on what can be seen as exceptional in real-world situations.

7.5.1 *Distribution of false discoveries*

In practice, one way to use Definition 7.7 for identifying exceptionality is to consider the extent to which subgroup models differ from the population model. In this case, we would like to identify models which are significantly different from the population model. The reason for shifting the focus to significantly different subgroups is that the true distribution of subgroups is unknown, and we therefore need to account for the error in the estimated model. Based on these ideas, the identification of exceptional subgroups is described next.

To determine how exceptional a subgroup G is, a sampling-based approach with the *distribution of false discoveries* (DFD, for short) [59, 112] is used. Suppose G has size $|G|$, then random subgroups of size $|G|$ are drawn without replacement from D, such that for each random subgroup its mismatch distance is computed. In order to compute the mismatch of each random subgroup, we fit a DBN on the random subgroup data and another DBN on its complement data. This sampling procedure approximates the distribution of mismatch distances that characterizes the mismatch of subgroups with size $|G|$.

By constructing a distribution of distances of random subgroups, we are able to assess how unusual the mismatch distance of a subgroup G is. In order to do so, we execute a hypothesis testing procedure as follows. By taking large enough number of sampled subgroups, the resulting distribution of random mismatch distances will be approximately Normal (see, e.g., [59, 112]). We can then compute a z-score for the mismatch of G, from which we can obtain a p-value. If the p-value of G is smaller than a significance level α, we conclude that G is an exceptional subgroup.

7.5.2 *Subgroup search*

In order to generate subgroups and test their exceptionality, we introduce a general search algorithm outlined in Algorithm 3. The central idea of Algorithm 3 is to specialize all exceptional subgroups that have been found so far, until there are no further exceptional subgroups to be specialized. The algorithm does not specialize subgroups considered as non-exceptional.

Algorithm 3 starts with $c = \emptyset$ as the current subgroup, i.e. the total population. By entering the outer loop, new candidate subgroups are generated by

specializing c with the addition of one descriptor that is not in the descriptor set of c (Line 8). For brevity sake, Line 8 in fact generates several subgroups, one for each value from the domain of the new descriptor. Then, each new candidate subgroup is tested for a minimal size σ and for exceptionality. If the candidate subgroup passes these tests, it is stored into the set E', which keeps the exceptional subgroups found so far. The new exceptional subgroup is also added to F, which stores the subgroups to further expand. Once the new exceptional subgroups have been processed, a subgroup to be further specialized is picked at random from F. While $F \neq \emptyset$, the whole specialization process is repeated.

Algorithm 3 Subgroup search

Input: D: a dataset of data points of the form $\{\mathbf{A}, \mathbf{X}^{(0)}, \mathbf{X}^{(1)}, \ldots\}$; σ: minimal size threshold; α: significance level for exceptionality test.
Output: E: set of exceptional subgroups.

1: $E \leftarrow \emptyset$
2: $F \leftarrow \emptyset$ // Subgroups to further expand
3: $c \leftarrow \emptyset$ // Current subgroup
4: cand_descs $\leftarrow \{A_1, \ldots, A_k\}$
5: **do**
6: $E' \leftarrow \emptyset$
7: **for all** $A_i \in$ get_cand_descriptors(c) **do**
8: $G \leftarrow c \cup \{A_i = a_i\}$, for each $a_i \in \mathrm{dom}(A_i)$
9: **if** check_size(G, D, σ) and exceptional(G, D, α) **then**
10: $E' \leftarrow E' \cup \{G\}$
// Add new exceptionals and select new one for expansion
11: $E \leftarrow E \cup E'$
12: $F \leftarrow F \cup E'$
13: $c \leftarrow$ select_random(F)
14: $F \leftarrow F - \{c\}$
15: **while** $F \neq \emptyset$
16: **return** E

7.5.3 *Exceptionality test*

Algorithm 3 makes use of an exceptionality test, which is detailed in Algorithm 4. Algorithm 4 does intensive computation as it learns subgroup models, calculates their mismatch distances, and calculates the DFDs. These steps are necessary to assess how unusual the mismatch of a particular subgroup is compared to random subgroups.

Algorithm 4 Exceptionality test

Input: G: a subgroup; D: a dataset of data points of the form $\{\mathbf{A}, \mathbf{X}^{(0)}, \mathbf{X}^{(1)}, \ldots\}$; α: significance level for exceptionality test.
Output: a Boolean value indicating whether G is exceptional.

$M_G \leftarrow \text{learn\_dbn}(S(G))$
$M_{\bar{G}} \leftarrow \text{learn\_dbn}(S(\bar{G}))$
$d \leftarrow \text{mismatch}(M_G, M_{\bar{G}})$
// Distribution of false discoveries
Sample subgroups from D with size $|G|$.
for all sampled subgroup H **do**
    $M_H \leftarrow \text{learn\_dbn}(S(H))$
    $M_{\bar{H}} \leftarrow \text{learn\_dbn}(S(\bar{H}))$
    $d_H \leftarrow \text{mismatch}(M_H, M_{\bar{H}})$
Calculate the mean and standard deviation from the set of distances d_H, and denote them by $\overline{x}$ and s respectively.
$z \leftarrow \dfrac{d - \overline{x}}{s}$ // z-score of the subgroup
Calculate the p-value corresponding to the z-score.
if p-value $< \alpha$ **then**
    return *true*
return *false*

7.5.4 *Search optimization*

The computation of DFDs is a costly step of the exceptionality test used by Algorithm 3. In order to evaluate the exceptionality of a subgroup G, we check whether a subgroup H with $|H| = |G|$ has been considered before during search. If so, we can reuse the previously computed DFD of H as the DFD of G, because the DFD is a function of the subgroup size. This can save substantial computation because in problems with several descriptor variables (the set $\mathbf{A}$), one would expect that some subgroups have the same size. We can take advantage of this fact by storing a list of sizes and a DFD for each size, so that a DFD is actually computed only when it is not found in this list.

By Proposition 7.2, the mismatch distance is symmetric. This means that if we ask whether a subgroup G with size $|G|$ is exceptional, we could equivalently ask whether the complementary subgroup (which has size $|D| - |G|$) is exceptional. This means that when we look up for a DFD in our table of stored DFDs, we can look up for DFDs associated to size $|G|$ *and* to DFDs associated to size $|D| - |G|$. This yields additional computational savings.

7.6 EXPERIMENTS WITH SIMULATED DATA

7.6.1 *Data*

We consider two simulation scenarios to assess the method by varying the set $\mathbf{X} = \{X_1, \ldots, X_n\}$, with X_i binary. In the first scenario, we use $n = 10$ variables inspired by previous research [112] which used Markov chains with 1,024 states. In the second scenario, we consider 100 times more MC states, requiring $n = \log_2 100 \cdot 1024 \simeq 17$ variables, allowing for a more comprehensive evaluation.

In order to build a dataset for a scenario, simulated data was generated from two ground truth DBNs based on the variables $\mathbf{X}$. The number of time points was 10 for both $n = 10$ and $n = 17$. The structure of each DBN was generated by uniformly sampling DAGs [122], while node parameters are sampled from Beta distributions.

The next step is to define the descriptor space. We defined a descriptor variable A_1 such that the sequences from one DBN were assigned to the subgroup $(A_1 = a_1^-)$ and the sequences from the other DBN to $(A_1 = a_1^+)$. The same amount of data was generated for these subgroups. We also added 5 binary descriptors $R_1, \ldots, R_5$ to act as noisy variables by randomly assigning the generated sequences to the noisy variables (with uniform probability).

Given a scenario, we now assign *ground truth labels* to unitary subgroups as follows:

- The subgroups (a_1^+) and (a_1^-) are seen as *positive instances*, as the sequences of each come from a single DBN, thus making these subgroups exceptional by definition.
- The subgroups described by R_i, such as $R_1 = r_1^+$ and $R_2 = r_2^-$, are seen as *negative instances*, as they correspond to random selections of sequences.

Based on the true and predicted labels, we measure how well we can identify exceptional subgroups (described by A_1) and non-exceptional subgroups (described by R_i). Further, by having only one descriptor for exceptional subgroups (A_1) and multiple ones for non-exceptional subgroups (R_i), it becomes more challenging to distinguish the two types of subgroups. This way we evaluate the robustness of the proposed algorithm.

Based on the described procedure, simulated data for a scenario consists of data points over the variables $\{A_1, R_1, \ldots, R_5, \mathbf{X}^{(0)}, \ldots, \mathbf{X}^{(9)}\}$. The whole simulation process, including the generation of ground truth models, was executed 10 times for better assessment of each scenario.

7.6.2 *Evaluation*

Algorithm 3 always generates unitary subgroups, which allows for evaluating the labeling done by the proposed method using several metrics. The AUC-ROC

	MC $n = 10$		DBN $n = 10$		MC $n = 17$		DBN $n = 17$	
Seq	**Pr**	**Rec**	**Pr**	**Rec**	**Pr**	**Rec**	**Pr**	**Rec**
10	0.6	0.6	0.82	1	0	0	0.9	1
20	0.87	1	0.95	1	0	0	0.9	1
40	0.85	1	0.87	1	0.15	0.2	0.9	1
60	0.92	1	0.87	1	0.53	0.6	0.89	1
80	0.92	1	0.83	1	0.75	0.85	0.88	1

Table 7.2: Precision (**Pr**) and recall (**Rec**) achieved by Markov chains and DBNs on simulated data. **Seq** = number of data sequences.

(area under the ROC curve) evaluates how the method separates the positive from the negative instances. We also compute *precision* and *recall* values, where precision is TP/(TP+FP) and recall is TP/(TP+FN) and TP, FP and FN denote the number of true positives, false positives, and false negatives.

Algorithm 3 also generates specialized subgroups if unitary exceptional subgroups are found. Specialized subgroups described by A_1 are also considered as exceptional. A subgroup such as (a_1^+, r_1^-) can be seen as a selection of half the sequences of subgroup (a_1^+), making the models of (a_1^+, r_1^-) and (a_1^+) similar. By opposition, specialized subgroups without A_1 are considered as non-exceptional. To facilitate comparisons, we evaluate unitary and specialized subgroups separately as the number of generated specialized subgroups can vary over different simulations. We used a size threshold $\sigma = 0.05$.

As a baseline, we consider Markov chains for representing the temporal targets instead of a DBN. In this case, the search algorithm is the same but the temporal targets are represented by a MC. To learn a MC, each variable $\mathbf{X}^{(t)}$ was mapped into a single variable $X'^{(t)}$ which has as domain the Cartesian product of the domains of $X_1, \ldots, X_n$. As a result, the state space of this MC can have up to 1,024 and 131,072 states for $n = 10$ and $n = 17$ respectively. Then, the temporal data of each sequence $\mathbf{X}^{(0)}, \mathbf{X}^{(1)}, \ldots$ was mapped into $X'^{(0)}, X'^{(1)}, \ldots$. This allows for an additional assessment of the DBN representation. To avoid zero probabilities, a Laplace smoothing [104] with $\lambda = 1$ is used in MC and DBN learning.

7.6.3 *Results*

Figure 7.1 shows the results based on simulated data for unitary subgroups. The results suggest that the DBN and the MC representation achieved good results with datasets of $n = 10$ target variables (or 1,024 MC states). However, substantial differences arose with $n = 17$ variables (or 131,072 MC states), a situation where DBNs were able to provide optimal AUC values even with the minimal amount of data, as opposed to MCs. In this case, MCs had to count on substantially larger amounts of data in order to provide comparable AUC values to those of DBNs. Table 7.2 shows the precision and recall of MCs and DBNs based on the threshold $\alpha = 0.05$ of Algorithm 4.

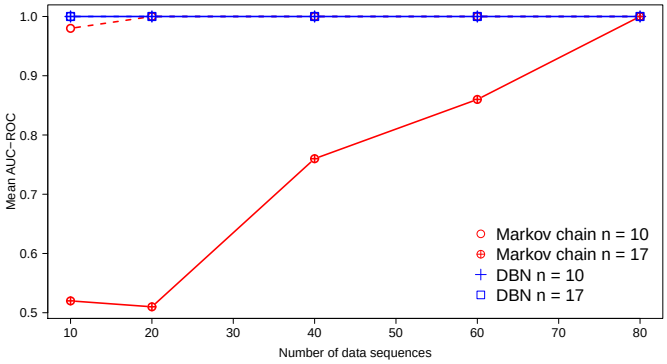


Figure 7.1: Effect of the amount of simulated data on the AUC-ROC of Markov chains and DBNs. Every sequence has 10 time points.

As previously discussed, specialized subgroups that include A_1 are supposed to be labeled as exceptional subgroups. Figure 7.2 shows the mean number of specialized subgroups which include A_1 and were labeled as exceptional. As the amount of data increases, the results show that more subgroups were produced by both the MC and DBN representations. However, it is clear that DBNs were able to capture significantly more specialized exceptional subgroups.

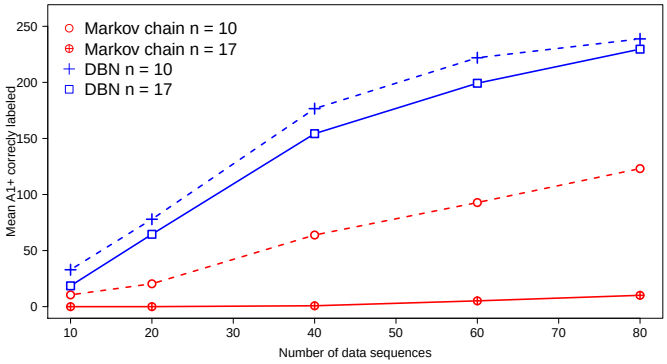


Figure 7.2: Mean number of specialized subgroups with A_1 which were labeled as exceptional (simulated data).

7.6.4 *Similar ground truth models*

Now we consider simulations where we control how similar the ground truth models are. This allows for a complementary evaluation of the search algorithm than that where we essentially varied the amount of data supplied to the algorithm. As before, two ground truth models are associated to the binary descriptor A_1.

In the following experiments, the second ground truth DBN was defined by copying the structure and parameters of the first DBN. For a variable X_i in the second DBN we have $p = P(X_i^{(0)} = x_i^- \mid \pi(x_i^{(0)}))$ and $p' = P(X_i^{(0)} = x_i^+ \mid \pi(x_i^{(0)}))$. These parameters were changed by picking at random a real number called *change* from the interval $[0, \min(\delta, 1-p)]$, with uniform probability, where $\delta \in [0,1]$ is the *maximal change threshold*. Next, we set $p = p + change$ and $p' = p' - change$. The lower the threshold δ, the more similar the DBNs are. It is straightforward to see that the modified p and p' values constitute a valid probability distribution.

Based on the previous results, we focus the analysis on DBNs in the remaining of this chapter. Figure 7.3 shows the AUC-ROC of simulations based on different maximal change thresholds. The results suggest that the search algorithm achieved better results with higher δ, which is expected because with more dissimilar ground truth models detecting exceptional behavior becomes more straightforward. On the other hand, the method made more mistakes under lower δ, particularly when there was little data, which can be seen as difficult situations for the method. In general, with larger amounts of data the method had better performance with any δ, which supports a behavior consistent with the previous experiments.

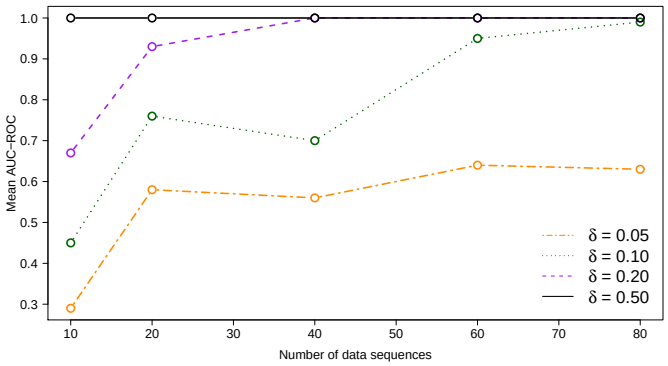


Figure 7.3: AUC-ROC achieved by DBNs on simulated data from different ground truth models. δ = maximal change threshold.

7.6.5 *Discussion*

Table 7.3 shows a fragment of subgroups from a simulation iteration using DBNs, together with their mismatch distances. This shows that the method is robust at identifying exceptional subgroups even when most of other subgroups are noisy subgroups. Moreover, the mismatch distances of exceptional subgroups are usually very different from those of non-exceptional subgroups.

The proposed mismatch score can be seen as a *data-based* score, as it is computed based on goodness-of-fit scores (the BIC score). By opposition, previous research

Subgroup	Size	z-score	p-value	Labels (I & T)	
(a_1^+)	0.50	195.8	$\simeq 0$	1	1
(a_1^-, r_2^-)	0.27	49.4	$\simeq 0$	1	1
(a_1^+, r_1^+, r_2^+)	0.11	15.1	$\simeq 0$	1	1
(r_2^-)	0.49	-1.2	0.22	0	0
(r_3^-)	0.49	0.5	0.64	0	0

Table 7.3: A simulation iteration based on DBNs ($n = 17$, 80 data sequences). **Size** = subgroup size normalized by $|D|$, **Labels (I&T)** = inferred and true labels respectively. The labels '1' and '0' indicate positive and negative instances respectively.

[112] for discovering exceptional MCs used a measure based on statistical distance between transition distributions. While structure learning is not required for MC learning, the number of parameters in DBNs is typically substantially lower due to its factorized representation. This is because the dimension of the transition matrices of MCs is prone to become very large even with a moderate number of target variables (e.g. $n = 17$).

As experiments have shown, this parameter issue makes the MC representation to scale poorly, particularly when n is larger and there is a reduced availability of data for model learning. Furthermore, the DBN-based search made substantially less mistakes in the simulations, which makes this representation suitable for TEMM.

7.7 DATA OF FUNDING APPLICATIONS

In order to evaluate the proposed TEMM method, we consider data from the *business process intelligence challenge* (BPIC18, for short) [56]. The BPIC18 dataset contains event log data of applications submitted to the European Union for direct payments for German farmers in 2015, 2016 and 2017. *The goal of applying TEMM to the BPIC18 data is to identify the subgroups in which the dynamic of events is exceptional*.

7.7.1 *Data*

Each application in the BPIC18 data is associated to descriptor variables (domain size) as follows: **Land Area** (437), **Department** (4), **Number of Parcels** (74), **Redistribution** (2), **Year of Submission** (3), **Success** (2), **Small Farmer** (2), and **Young Farmer** (2). Applications are also associated to *events* related to workflow activities, where an event is described by the multinomial variables (domain size): **Activity** (41), **Doctype** (8), **Subprocess** (8). Each application is associated to one or more events, which are the temporal targets of the data. Hence, the ith data point of this dataset has the form $\{\text{Land Area}, \ldots, \text{Young Farmer}, \text{Activity}^{(0:m_i)}, \ldots, \text{Subprocess}^{(0:m_i)}\}$, where m_i is its last time point.

The BPIC18 dataset has 4,800 applications randomly selected from the original dataset, with an equal number of applications per year. The dataset considered for the experiments has 275,226 events in total (mean [StDv] length of each application: 57.3 [49.5] events).

7.7.2 *Discovered subgroups*

Table 7.4 shows the exceptional and non-exceptional subgroups that were discovered from the BPIC18 data based on a minimal size $\sigma = 0.05$. The results suggest that the most exceptional subgroups are unitary and described by a particular year, be it 2015, 2016 or 2017. This might suggest that significant changes took place in application processing between different years, such as changes in application structure, time spent in application tasks, funding policies, etc. Regardless of the year, each department has its own dynamics, as all unitary subgroups (Department) were exceptional. However, their the exceptionality was not as strong as that of (Year) subgroups.

As Table 7.4 shows, unitary subgroups of (Young Farmer) were not exceptional, which suggests that the exceptionality of subgroups as $(\text{Year} = 2017 \wedge \text{Young Farmer}^{-})$ is only caused by other attributes. Due to the large size of $(\text{young.farmer}^{-})$, we conjecture that some specialized subgroups of (Young Farmer) have distributions similar to their generalized subgroups without (Young Farmer), which would make such specialized subgroups redundant.

7.7.3 *Validation*

The BPIC18 data provider [56] claims that the underlying process changed between years due to changes implemented in the structure of the application procedure. This is evidence that supports the exceptional subgroups found in this chapter described by (Year), as shown in Table 7.4.

Such discovered exceptional subgroups are also in line with previous research [135] applied to this dataset, which was able to identify concept drifts precisely between each year of the data. Other research [174] has analyzed how the workflow of applications submitted in different years has changed, also suggesting that differences exist in the workflow structure between years.

Differently than the other analyses from the literature on the BPIC18 data, the method proposed in this chapter can be seen as a principled one due to its automated nature. However, the discussed validation of the subgroups found should be seen as a partial validation, as the true exceptional subgroups of real-world data are usually unknown.

7.8 CONCLUSIONS

In this chapter, we proposed TEMM, a generalization of EMM to allow for the representation of multiple and temporal targets. We proposed a method able to

Exceptional subgroups	**Size**	**z-score**
year = 2017	0.34	773.6
year = 2015	0.35	524.1
year = 2016	0.30	479.0
department = e7	0.30	23.4
department = d4	0.16	21.3
department = 4e	0.30	13.1
department = 6b	0.24	11.3
number_parcels = 2	0.06	7.2
year = 2017 $\wedge$ young.farmer$^-$	0.31	385.0
year = 2015 $\wedge$ young.farmer$^-$	0.32	363.7
department = e7 $\wedge$ year = 2017	0.10	166.6
department = 6b $\wedge$ year = 2017	0.09	110.9
department = 6b $\wedge$ year = 2016	0.07	106.7
department = 6b $\wedge$ young.farmer$^-$ $\wedge$ year = 2016	0.06	147.6
department = e7 $\wedge$ young.farmer$^-$ $\wedge$ year = 2017	0.09	128.2
department = 4e $\wedge$ young.farmer$^-$ $\wedge$ year = 2017	0.09	124.9
department = 6b $\wedge$ young.farmer$^-$ $\wedge$ year = 2017	0.08	118.3
department = e7 $\wedge$ young.farmer$^-$ $\wedge$ year = 2016	0.08	69.6
Non-exceptional subgroups	**Size**	**z-score**
young.farmer$^-$	0.91	1.7
young.farmer$^+$	0.09	1.3
number_parcels = 3	0.06	0.9
department = e7 $\wedge$ year = 2015	0.11	0.2
department = 4e $\wedge$ year = 2015	0.10	-1.6

Table 7.4: Exceptional (34) and non-exceptional (5) subgroups from the BPIC18 dataset. For better visualization, the 5 most specialized subgroups are shown. **Size** = subgroup size normalized by $|D|$. All p-values < 0.001 (exceptional subgroups) and ≥ 0.05 (non-exceptional subgroups).

identify exceptional DBNs from temporal data, which allows for an intuitive and sound model class for TEMM.

The proposed TEMM method was empirically evaluated on simulated data and a process data based on funding applications, showing that the identifiability of the method in different scenarios is robust. Our method was able to discover exceptional subgroups from the funding data in accordance to previous research, as well other, yet less exceptional subgroups. Furthermore, our approach solved this practical problem in a more principled manner.

As future work, we would like to better explain why models are considered as exceptional, e.g., by looking at relevant structural or numerical parameters of the DBNs. We also wish to summarize exceptional subgroups that might reflect the same DBN distribution, e.g., by merging exceptional subgroups during search or post-processing. Moreover, by investigating the relation between subgroup size and the mismatch distance, the search mechanism could be further optimized.

8

DISCUSSION

This thesis dealt with the discovery of the underlying structure of temporal processes. The underlying structure of a process can be captured by different mathematical representations, depending on which aspects one wishes to focus on. Yet, the task of choosing suitable models to represent a particular real-world problem is challenging.

In this work, we advocate that the proposed methods based on probabilistic graphical models are advantageous when solving several real-world problems. Just as PGMs, the methods introduced in this work can work well under situations of varied data scarcity and are often interpretable. One advantage of being interpretable is that insight can often be obtained from such models without a lot of barriers. Although models such as deep neural networks have been recently very successful at tasks such as supervised learning, they tend to require higher amounts of data, large amounts of computational resources (e.g., computing time) and are considerably less interpretable, characteristics which go in an opposite direction to PGMs.

8.1 CONTRIBUTIONS

The central theme covered in this work is the different viewpoints on dynamics of temporal processes, which we summarize in the following sections.

8.1.1 *Asymmetry in models*

Hidden Markov models have been used for providing a viewpoint of processes based on latent variables. In order to increase the problem insight and model fit that one can obtain from HMMs, we proposed asymmetric HMMs (HMM-As, for short) in Chapter 3.

HMM-As allow for more expressive representation of observations by capturing distribution asymmetries (also known as local structure). As a result, HMM-As often need fewer latent states to achieve model fit at least as good as that of other HMM models. The parsimonious representation of HMM-As is also valuable in model interpretation. Due to the large number of variations of existing HMMs, it

is often difficult to decide which model class would be suitable for the problem at hand. The flexibility of the HMM-A representation reduces the need for deciding which class of symmetric and other asymmetric HMMs one should use. In Chapter 3, we demonstrated the aforementioned advantages of HMM-As by means of simulated and real-world data from several domains.

8.1.2 *Generation of hypotheses on processes*

We proposed in Chapter 4 a method that helps the selection of hypotheses to further investigate about disease processes. This method is semi-automatic and is based on structured HMMs, particularly with the goal of generating insight on disease processes. The formulation of outcomes is aided by means of *state reachability* that one can build by looking at model aspects such as state probabilities at different time points. The state reachability notion was shown to help understanding patient trajectories in a compact way.

A case study of psychotic depression was considered in Chapter 4, for which hypotheses were generated. One of the main results discussed is that patients undergoing psychotic depression treatment showed to be sensitive to treatment based on their initial psychotic symptoms. By using this methodology, new knowledge about this mental disorder was acquired, which potentially helps doctors to prescribe more efficient medication in the future.

8.1.3 *Capturing hidden (non-observed) aspects of processes*

Besides the independence structure of processes given by asymmetric HMMs (Chapter 3), we investigated in Chapter 5 how to gain insight in health care data by means of latent-variable modeling. In this chapter we introduced the notion of cluster of hidden states based on hidden Markov models. Clusters of states can be learned from datasets with a single event produced at each instant.

Based on health care data from Dutch practices, clusters of latent states were learned. The clusters were shown to provide additional characterization to the latent states by suggesting that states from each cluster are correlated to different patient severity. Ultimately, the results of Chapter 5 also allow for gaining insight on multimorbidity by analyzing the clusters of states learned for different disease codes.

8.1.4 *Taking into account the size of datasets*

Learning insightful models from small datasets is known to be a challenging problem. In this work, we proposed new methodologies and model classes that showed to be useful in real-world situations of the limited availability of data. In Chapter 6, we proposed partitioned dynamic Bayesian networks for representing dynamic Bayesian networks with regime change over time. A heuristic was

proposed for identifying regime cut-offs of PDBNs in a parsimonious way, which favors the situation of small datasets.

Experiments based on simulations showed that PDBNs learned by the proposed heuristic search dealt well with different situations, in particular with small datasets generated by different underlying models. PDBNs learned from psychotic depression treatment data provided a better model fit than DBNs and more insight on the interaction between psychotic and depressive features. Different viewpoints on disease dynamics for psychotic depression treatment were provided in Chapters 4 and 6 by investigating different variables and problem aspects.

8.1.5 *Temporal subgroups*

A new problem called temporal exceptional model mining (TEMM, for short) was defined in Chapter 7. TEMM aims to discover exceptional behavior associated to subsets of the data that can be described by a configuration of variables.

We proposed a method that allows for the discovery of exceptional dynamic Bayesian networks by means of a distance measure and a search algorithm. The proposed temporal representation allows for more accurate retrieval of exceptional subgroups than that based on simpler temporal models such as Markov chains. The method for identifying exceptional DBNs was evaluated based on real-world process data and was able to discover exceptional subgroups in a principled fashion. The results were validated in comparison to previous research, based on different techniques.

8.2 FUTURE WORK

In this section, we discuss limitations and directions for future work that we believe might be relevant. We also discuss potential approaches that could be relevant for achieving these goals.

8.2.1 *Asymmetry in models*

The number of states of HMM-As was selected by a trial and error approach. It is worthwhile to investigate more principled ways to make this selection. One alternative is to use infinite HMMs [8] to automatically determine the number of hidden states. Another alternative is to predict the number of states by means of Bayesian optimization [156], where the performance (e.g. the goodness of fit) of each state could be seen as a complex black-box function.

8.2.2 *Generation of hypotheses on processes*

We would like to apply the hypothesis generation methodology proposed in Chapter 4 in a more *automatic* fashion. This means automatizing the selection of

baseline and target states, the computation of state reachabilities, and so on. This is already possible given the proposed method, which would likely result in a set of candidate hypotheses to further investigate.

By applying the method of Chapter 4 to other datasets, it would be possible to evaluate the effectiveness of automatic generation of hypothesis. The advantage of using multiple datasets is a more effective evaluation of the effect of different definitions for selecting baseline and target states, including those definitions proposed in Chapter 4.

One new challenge that would likely arise by automatizing hypothesis generation is how to properly assess different hypotheses that would be generated. To that end, one could benefit from currently available electronic health records [30, 142], which might lead to additional sources of information, such as clinical notes written by medical doctors (often in natural language). Such data could perhaps give direction to which hypotheses could be promising to be investigated, e.g., by assigning some kind of utility to the generated hypothesis.

8.2.3 *Capturing hidden (non-observed) aspects of processes*

In Chapter 5, the analysis of clusters based on a medical outcome (in this case, disease counts) was carried out after the clusters of states were identified. We would like to integrate medical outcomes as part of the models, so that the relationship with the clusters of states and outcome measure could be direct and more general, which would allow one to look at such relationship by different angles. One could consider doing this by integrating such outcomes during model learning, e.g., by means of covariates along the lines of Input-Output HMMs [11].

A small number of diagnosis variables was considered when learning clusters of states based on the case study of Chapter 5. It would be interesting to look at a larger set of variables and the resulting clusters, which seems a natural extension to the presented method since health care data often have hundreds or more diagnosis codes.

To make multimorbidity analysis more effective, we would like to consider data representations where multiple diseases occur at the same time. This would, however, result in a significant departure from the modeling assumptions considered of Chapter 5, as the observation space would likely have several active variables, which could create the need for a different notion of cluster of states. The advantage, however, is that multimorbidity could be analyzed in a more direct way.

8.2.4 *Taking into account the size of datasets*

We would like to extend PDBNs to make it possible to model recurrent regimes over time, such that the identified regimes could be seen as states similarly to HMM states. This could make PDBNs more compact and more explainable

models. Related research that considers recurrent regimes includes, e.g., gated networks [10] and DBNs with an HMM-based dependence structure [78].

We also would like to broaden the evaluation of PDBNs by comparing them to other non-homogeneous models, which are usually based on different assumptions. This would allow for evaluating different classes of models under different assumptions and how they perform when some (or all) assumptions are not met.

8.2.5 *Temporal subgroups*

In TEMM, no relationship between discovered subgroups is explicitly computed. Nevertheless, one can argue that some kind of relationship between the discovered exceptional subgroups might exist. One simple example is when multiple subgroups represent (approximately) the same exceptional behavior, i.e., they deviate from the population in (approximately) the same way.

By identifying subgroups that are exceptional and yet similar, not only redundancies could be reduced, but also more insight about the problem would be obtained as to how exceptional behavior might occur. Redundancies could be eliminated by rejecting specialized subgroups that are similar to more general subgroups (see, e.g., [112]). Instead of rejecting subgroups, an alternative is to introduce merge and split operations in the search algorithm. In that case, the language for expressing subgroups could be extended to represent other description patterns beyond pairs of attribute-value, which could allow for a more general understanding between description patterns and exceptional behavior.

Another research direction that might be worthwhile to explore in TEMM is what makes a model an exceptional model. While this might be more or less evident when one deals with exceptional models based on just a few variables (as in standard subgroup discovery, for example), this is no longer the case for more complex models such as dynamic Bayesian networks. For DBNs, one could be interested in knowing whether particular pieces of the model structure or particular parameters are relevant for explaining why a DBN is considered as exceptional. Sensitivity analysis [34, 72] might be of help for pursuing this research direction.

The computation of distributions of false discoveries based on DBNs is expensive. We would like to investigate whether further optimizations can be used to reduce these computations. One idea is to try to predict new distributions of false discoveries based on previously computed ones, e.g., by predicting its parameters such as the mean and standard deviation. One method that might help is Bayesian optimization [156, 160], as obtaining the distributions of false discoveries can be seen as evaluating an expensive black-box function.

BIBLIOGRAPHY

[1] W. van der Aalst. *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. Springer Publishing Company, Incorporated, 2011. ISBN: 3642193447, 9783642193446.

[2] E. Ahlqvist et al. "Novel subgroups of adult-onset diabetes and their association with outcomes : a data-driven cluster analysis of six variables." In: *The Lancet Diabetes and Endocrinology* 6.5 (2018), pp. 361–369. DOI: 10.1016/S2213-8587(18)30051-2.

[3] K. J. Anstey and S. M. Hofer. "Longitudinal Designs, Methods and Analysis in Psychiatric Research." In: *Australian & New Zealand Journal of Psychiatry* 38.3 (2004). PMID: 14961925, pp. 93–104. DOI: 10.1080/j.1440-1614.2004.01343.x.

[4] M. Atzmueller, S. Doerfel, and F. Mitzlaff. "Description-oriented community detection using exhaustive subgroup discovery." In: *Information Sciences* 329 (2016). Special issue on Discovery Science, pp. 965–984. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2015.05.008.

[5] L. M. Barclay, R. A. Collazo, J. Q. Smith, P. A. Thwaites, and A. E. Nicholson. "The dynamic chain event graph." In: *Electron. J. Statist.* 9.2 (2015), pp. 2130–2169.

[6] K. Barnett, S. W. Mercer, M. Norbury, G. Watt, S. Wyke, and B. Guthrie. "Epidemiology of multimorbidity and implications for health care, research, and medical education: a cross-sectional study." In: *The Lancet* 380 (July 2012), pp. 37–43. ISSN: 0140-6736. DOI: 10.1016/S0140-6736(12)60240-2.

[7] M. Bartlett and J. Cussens. "Integer Linear Programming for the Bayesian network structure learning problem." In: *Artificial Intelligence* 244 (2017). Combining Constraint Solving with Mining and Learning, pp. 258–271. ISSN: 0004-3702. DOI: https://doi.org/10.1016/j.artint.2015.03.003.

[8] M. J. Beal, Z. Ghahramani, and C. E. Rasmussen. "The infinite hidden Markov model." In: *Advances in neural information processing systems*. 2002, pp. 577–584.

[9] A. Bendimerad, M. Plantevit, and C. Robardet. "Mining exceptional closed patterns in attributed graphs." In: *Knowledge and Information Systems* 56.1 (2018), pp. 1–25. ISSN: 0219-3116. DOI: 10.1007/s10115-017-1109-2.

[10] M. Bendtsen. "Regime Aware Learning." In: *Proceedings of the Eighth International Conference on Probabilistic Graphical Models*. Ed. by A. Antonucci, G. Corani, and C. P. Campos. 2016, pp. 1–12.

[11] Y. Bengio and P. Frasconi. "An Input Output HMM Architecture." In: *Advances in Neural Information Processing Systems*. Ed. by G. Tesauro, D. Touretzky, and T. Leen. Vol. 7. The MIT Press, 1995, pp. 427–434.

[12] J. Bilmes. "Dynamic Bayesian multinets." In: *Proc. of the Sixteenth conference on Uncertainty in Artificial Intelligence*. 2000, pp. 38–45.

[13] J. Bilmes. "What HMMs Can Do." In: *IEICE - Trans. Inf. Syst.* E89-D.3 (Mar. 2006), pp. 869–891.

[14] J. Bilmes. *A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models*. Tech. rep. TR-97-021. International Computer Science Institute, Berkeley, 1998, p. 126.

[15] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006. ISBN: 0387310738.

[16] C. van Borkulo, L. Boschloo, D. Borsboom, B. Penninx, L. Waldorp, and R. Schoevers. "Association of symptom network structure with the course of depression." In: *JAMA Psychiatry* 72.12 (2015), pp. 1219–1226. DOI: 10.1001/jamapsychiatry.2015.2079.

[17] D. Borsboom, G. J Mellenbergh, and J. Heerden. "The Theoretical Status of Latent Variables." In: *Psychological review* 110 (May 2003), pp. 203–19. DOI: 10.1037/0033-295X.110.2.203.

[18] G. Bosc, J.-F. Boulicaut, C. Raïssi, and M. Kaytoue. "Anytime discovery of a diverse set of patterns with Monte Carlo tree search." In: *Data Mining and Knowledge Discovery* 32.3 (2018), pp. 604–650. ISSN: 1573-756X. DOI: 10.1007/s10618-017-0547-5.

[19] C. Boutilier, N. Friedman, M. Goldszmidt, and D. Koller. "Context-specific Independence in Bayesian Networks." In: *Proc. of the 20th Intl. Conf. on Uncertainty in Artificial Intelligence*. 1996, pp. 115–123.

[20] L. Breiman. "Bagging predictors." In: *Machine learning* 24.2 (1996), pp. 123–140.

[21] G. Van den Broeck, K. Mohan, A. Choi, A. Darwiche, and J. Pearl. "Efficient Algorithms for Bayesian Network Parameter Learning from Incomplete Data." In: *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*. UAI'15. Amsterdam, Netherlands: AUAI Press, 2015, pp. 161–170. ISBN: 978-0-9966431-0-8.

[22] M. L. P. Bueno, A. Hommersom, and P. J. F. Lucas. "Exceptional model mining using dynamic Bayesian networks." In: *under review*. 2020.

[23] M. L. P. Bueno, A. Hommersom, P. J. F. Lucas, S. Verwer, and A. Linard. "Learning Complex Uncertain States Changes via Asymmetric Hidden Markov Models: an Industrial Case." In: *Proceedings of the Eighth International Conference on Probabilistic Graphical Models*. Ed. by A. Antonucci, G. Corani, and C. P. Campos. Vol. 52. Proceedings of Machine Learning Research. Lugano, Switzerland: PMLR, 2016, pp. 50–61.

[24] M. L. P. Bueno, A. Hommersom, P. J. F. Lucas, M. Lappenschaar, and J. G. Janzing. "Understanding disease processes by partitioned dynamic Bayesian networks." In: *Journal of Biomedical Informatics* 61 (2016), pp. 283–297. ISSN: 1532-0464. DOI: https://doi.org/10.1016/j.jbi.2016.05.003.

[25] M. L. P. Bueno, A. Hommersom, P. J. F. Lucas, and A. Linard. "Asymmetric hidden Markov models." In: *International Journal of Approximate Reasoning* 88 (2017), pp. 169–191. ISSN: 0888-613X. DOI: https://doi.org/10.1016/j.ijar.2017.05.011.

[26] M. L. P. Bueno, A. Hommersom, P. J. F. Lucas, M. Lobo, and P. P. Rodrigues. "Modeling the Dynamics of Multiple Disease Occurrence by Latent States." In: *Scalable Uncertainty Management - 12th International Conference, SUM*. 2018, pp. 93–107. DOI: 10.1007/978-3-030-00461-3_7.

[27] M. L. P. Bueno, A. Hommersom, P. J. F. Lucas, and J. Janzing. "A Data-Driven Exploration of Hypotheses on Disease Dynamics." In: *16th Conference on Artificial Intelligence in Medicine, AIME*. 2019, pp. 1–10. DOI: 10.1007/978-3-030-21642-9_23.

[28] M. L. P. Bueno, A. Hommersom, P. J. F. Lucas, and J. Janzing. "A probabilistic framework for predicting disease dynamics: A case study of psychotic depression." In: *Journal of Biomedical Informatics* 95 (2019), p. 103232. ISSN: 1532-0464. DOI: https://doi.org/10.1016/j.jbi.2019.103232.

[29] J. Burge, T. Lane, H. Link, S. Qiu, and V. P. Clark. "Discrete dynamic Bayesian network analysis of fMRI data." In: *Human Brain Mapping* 30.1 (2009), pp. 122–137.

[30] R. J. Byrd, S. R. Steinhubl, J. Sun, S. Ebadollahi, and W. F. Stewart. "Automatic identification of heart failure diagnostic criteria, using text analysis of clinical notes from electronic health records." In: *International Journal of Medical Informatics* 83.12 (2014), pp. 983–992. ISSN: 1386-5056. DOI: https://doi.org/10.1016/j.ijmedinf.2012.12.005.

[31] C. P. de Campos and Q. Ji. "Efficient Structure Learning of Bayesian Networks Using Constraints." In: *J. Mach. Learn. Res.* 12 (July 2011), pp. 663–689. ISSN: 1532-4435.

[32] A. Cano, M. Gémez-Olmedo, and S. Moral. "Approximate inference in Bayesian networks using binary probability trees." In: *International Journal of Approximate Reasoning* 52.1 (2011). Tenth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2009), pp. 49–62. ISSN: 0888-613X. DOI: https://doi.org/10.1016/j.ijar.2010.05.006.

[33] R. Castelo and T. Kocka. "On Inclusion-driven Learning of Bayesian Networks." In: *J. Mach. Learn. Res.* 4 (Dec. 2003), pp. 527–574.

[34] E. Castillo, J. M. Gutierrez, and A. S. Hadi. "Sensitivity Analysis in Discrete Bayesian Networks." In: *Trans. Sys. Man Cyber. Part A* 27.4 (July 1997), pp. 412–423. ISSN: 1083-4427.

[35] G. Celeux and J.-B. Durand. "Selecting hidden Markov model state number with cross-validated likelihood." In: *Computational Statistics* 23.4 (2008), pp. 541–564. ISSN: 1613-9658. DOI: 10.1007/s00180-007-0097-1.

[36] Y.-S. Chang, C.-T. Fan, W.-T. Lo, W.-C. Hung, and S.-M. Yuan. "Mobile cloud-based depression diagnosis using an ontology and a Bayesian network." In: *Future Generation Computer Systems* 43–44 (2015), pp. 87 –98.

[37] T. Charitos and L. C. van der Gaag. "Sensitivity Analysis for Threshold Decision Making with Dynamic Networks." In: *UAI '06, Proceedings of the 22nd Conference in Uncertainty in Artificial Intelligence, Cambridge, MA, USA, July 13-16*. 2006.

[38] T. Charitos, L. C. van der Gaag, S. Visscher, K. A. Schurink, and P. J. F. Lucas. "A dynamic Bayesian network for diagnosing ventilator-associated pneumonia in ICU patients." In: *Expert Systems with Applications* 36.2, Part 1 (2009), pp. 1249–1258.

[39] M. Chavira and A. Darwiche. "On probabilistic inference by weighted model counting." In: *Artificial Intelligence* 172.6 (2008), pp. 772–799. ISSN: 0004-3702.

[40] J. Cheng, R. Greiner, J. Kelly, D. Bell, and W. Liu. "Learning Bayesian networks from data: An information-theory based approach." In: *Artificial Intelligence* 137 (2002), pp. 43–90.

[41] J.-P. Chevrolat, J.-L. Golmard, S. Ammar, R. Jouvent, and J.-F. Boisvieux. "Modelling behavioral syndromes using Bayesian networks." In: *Artificial Intelligence in Medicine* 14 (3 1998), pp. 259–277.

[42] D. M. Chickering. "Learning Bayesian Networks is NP-Complete." In: *Learning from Data: Artificial Intelligence and Statistics V*. Ed. by D. Fisher and H.-J. Lenz. New York, NY: Springer New York, 1996, pp. 121–130. ISBN: 978-1-4612-2404-4. DOI: 10.1007/978-1-4612-2404-4_12.

[43] D. M. Chickering, D. Heckerman, and C. Meek. "Large-Sample Learning of Bayesian Networks is NP-Hard." In: *J. Mach. Learn. Res.* 5 (Dec. 2004), pp. 1287–1330. ISSN: 1532-4435.

[44] G. F. Cooper and E. Herskovits. "A Bayesian method for the induction of probabilistic networks from data." In: *Machine Learning* 9.4 (1992), pp. 309–347. DOI: 10.1007/BF00994110.

[45] M. J. Côté and W. E. Stein. "A stochastic model for a visit to the doctor's office." In: *Mathematical and Computer Modelling* 45.3 (2007), pp. 309–323. ISSN: 0895-7177. DOI: https://doi.org/10.1016/j.mcm.2006.03.022.

[46] T. M. Cover and J. A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, 2006.

[47] R. G. Cowell, S. L. Lauritzen, A. P. David, and D. J. Spiegelhalter. *Probabilistic Networks and Expert Systems*. Ed. by V. Nair, J. Lawless, and M. Jordan. 1st. Berlin, Heidelberg: Springer-Verlag, 1999. ISBN: 0387987673.

[48] A. O. J. Cramer, L. J. Waldorp, H. L. J. van der Maas, and D. Borsboom. "Comorbidity: A network perspective." In: *Behavioral and Brain Sciences* 33.2-3 (2010), pp. 137–150. DOI: 10.1017/s0140525x09991567.

[49] D.-I. Curiac, G. Vasile, O. Banias, C. Volosencu, and A. Albu. "Bayesian network model for diagnosis of psychiatric diseases." In: *Information Technology Interfaces, 2009. ITI '09. Proceedings of the ITI 2009 31st International Conference on*. 2009, pp. 61–66.

[50] J. Cussens, M. Järvisalo, J. H. Korhonen, and M. Bartlett. "Bayesian Network Structure Learning with Integer Programming: Polytopes, Facets and Complexity." In: *J. Artif. Intell. Res.* 58 (2017), pp. 185–229.

[51] R. Daly, Q. Shen, and S. Aitken. "Review: Learning Bayesian Networks: Approaches and Issues." In: *Knowl. Eng. Rev.* 26.2 (May 2011), pp. 99–157. ISSN: 0269-8889.

[52] M. H. DeGroot and M. J. Schervish. *Probability and Statistics*. 3rd. Pearson, 2011.

[53] A. P. Dempster, N. M. Laird, and D. B. Rubin. "Maximum likelihood from incomplete data via the EM algorithm." In: *J. of The Royal Statistical Soc., Series B* 39.1 (1977), pp. 1–38.

[54] F. Dondelinger, S. Lèbre, and D. Husmeier. "Non-homogeneous dynamic Bayesian networks with Bayesian regularization for inferring gene regulatory networks with gradually time-varying structure." English. In: *Machine Learning* 90.2 (2013), pp. 191–230.

[55] B. van Dongen; *BPI Challenge 2014: Interaction details*. 2014. DOI: 10.4121/uuid:3d5ae0ce-198c-4b5c-b0f9-60d3035d07bf.

[56] B. van Dongen and F. Borchert. *BPI Challenge 2018*. en. 2018. DOI: 10.4121/UUID:3301445F-95E8-4FF0-98A4-901F1F204972. URL: https://data.4tu.nl/repository/uuid:3301445f-95e8-4ff0-98a4-901f1f204972.

[57] W. Duivesteijn, A. Knobbe, A. Feelders, and M. van Leeuwen. "Subgroup Discovery Meets Bayesian Networks – An Exceptional Model Mining Approach." In: *2010 IEEE International Conference on Data Mining*. 2010, pp. 158–167. DOI: 10.1109/ICDM.2010.53.

[58] W. Duivesteijn, A. J. Feelders, and A. Knobbe. "Exceptional Model Mining." In: *Data Min. Knowl. Discov.* 30.1 (Jan. 2016), pp. 47–98. ISSN: 1384-5810. DOI: 10.1007/s10618-015-0403-4.

[59] W. Duivesteijn and A. Knobbe. "Exploiting False Discoveries – Statistical Validation of Patterns and Quality Measures in Subgroup Discovery." In: *Proceedings of the 2011 IEEE 11th International Conference on Data Mining*. ICDM '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 151–160. ISBN: 978-0-7695-4408-3.

[60] S. R. Eddy. "Accelerated Profile HMM Searches." In: *PLOS Computational Biology* 7.10 (Oct. 2011), pp. 1–16.

[61] M. Fani Sani, W. van der Aalst, A. Bolt, and J. García-Algarra. "Subgroup Discovery in Process Mining." In: *Business Information Systems*. Ed. by W. Abramowicz. Cham: Springer International Publishing, 2017, pp. 237–252. ISBN: 978-3-319-59336-4.

[62] S. Fine, Y. Singer, and N. Tishby. "The Hierarchical Hidden Markov Model: Analysis and Applications." In: *Mach. Learn.* 32.1 (July 1998), pp. 41–62.

[63] K. Frank, M. Röckl, M. J. V. Nadales, P. Robertson, and T. Pfeifer. "Comparison of exact static and dynamic Bayesian context inference methods for activity recognition." In: *2010 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*. 2010, pp. 189–195. DOI: 10.1109/PERCOMW.2010.5470671.

[64] D. Freitag and A. McCallum. "Information Extraction with HMM Structures Learned by Stochastic Optimization." In: *Proc. of the 17th AAAI and 20th IAAI*. AAAI Press, 2000, pp. 584–589.

[65] N. Friedman. "Learning Belief Networks in the Presence of Missing Values and Hidden Variables." In: *Proc. of the 14th ICML*. ICML '97. San Francisco, USA: Morgan Kaufmann Publishers Inc., 1997, pp. 125–133.

[66] N. Friedman and M. Goldszmidt. "Learning Bayesian Networks with Local Structure." In: *Proc. of the 20th UAI*. UAI'96. Portland, OR: Morgan Kaufmann Publishers Inc., 1996, pp. 252–262.

[67] N. Friedman. "The Bayesian Structural EM Algorithm." In: *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*. UAI'98. Madison, Wisconsin: Morgan Kaufmann Publishers Inc., 1998, pp. 129–138. ISBN: 1-55860-555-X.

[68] N. Friedman, K. Murphy, and S. Russell. "Learning the Structure of Dynamic Probabilistic Networks." In: *Proc. of the 14th Conference on Uncertainty in Artificial Intelligence*. UAI'98. Madison, Wisconsin: Morgan Kaufmann Publishers Inc., 1998, pp. 139–147.

[69] B. A. Frigyik, A. Kapila, and M. R. Gupta. *Introduction to the Dirichlet Distribution and Related Processes.* Technical report UWEETR-2010-0006, Department of Electrical Engineering, University of Washington. 2010.

[70] L. van der Gaag. "Bayesian Belief Networks: Odds and Ends." In: *The Computer Journal* 39.2 (Jan. 1996), pp. 97–113. ISSN: 0010-4620. DOI: 10.1093/comjnl/39.2.97.

[71] L. van der Gaag, S. Renooij, C. Witteman, B. Aleman, and B. Taal. "How to Elicit Many Probabilities." In: *Proceedings of the Fifteenth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-99)*. San Francisco, CA: Morgan Kaufmann, 1999, pp. 647–654.

[72] L. C. van der Gaag, S. Renooij, and V. M. Coupé. "Sensitivity Analysis of Probabilistic Networks." In: *Advances in Probabilistic Graphical Models*. Ed. by P. Lucas, J. A. Gámez, and A. Salmerón. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 103–124.

[73] D. Geiger and D. Heckerman. "Knowledge representation and inference in similarity networks and Bayesian multinets." In: *Artificial Intelligence* 82.1 (1996), pp. 45–74.

[74] M. A. van Gerven, B. G. Taal, and P. J. F. Lucas. "Dynamic Bayesian networks as prognostic models for clinical patient management." In: *Journal of Biomedical Informatics* 41.4 (2008), pp. 515–529.

[75] Z. Ghahramani. "An Introduction to Hidden Markov Models and Bayesian Networks." In: *International Journal of Pattern Recognition and Artificial Intelligence* (2001), pp. 9–42.

[76] Z. Ghahramani and M. Jordan. "Factorial Hidden Markov Models." In: *Machine Learning* 29.2 (1997), pp. 245–273.

[77] F. Glover and M. Laguna. *Tabu Search*. Norwell, MA, USA: Kluwer Academic Publishers, 1997. ISBN: 079239965X.

[78] M. Grzegorczyk. "A non-homogeneous dynamic Bayesian network with a hidden Markov model dependency structure among the temporal data points." In: *Machine Learning* 102.2 (2016), pp. 155–207. ISSN: 1573-0565. DOI: 10.1007/s10994-015-5503-2.

[79] M. Grzegorczyk and D. Husmeier. "Non-stationary continuous dynamic Bayesian networks." In: *Advances in Neural Information Processing Systems 22*. Ed. by Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta. 2009, pp. 682–690.

[80] D. Gunning. *Explainable Artificial Intelligence (XAI)*. Ed. by DARPA. DARPA, 2016. URL: http://www.darpa.mil/program/explainable-artificial-intelligence.

[81] M. Hamilton. "A RATING SCALE FOR DEPRESSION." In: *Journal of Neurology, Neurosurgery & Psychiatry* 23.1 (1960), pp. 56–62. ISSN: 0022-3050. DOI: 10.1136/jnnp.23.1.56.

[82] C. A. Hammerschmidt, S. Verwer, Q. Lin, and R. State. "Interpreting Finite Automata for Sequential Data." In: *Interpretable Mach. Learn. for Compl. Sys.: NIPS 2016 workshop proc.* 2016.

[83] M. Hauskrecht, I. Batal, M. Valko, S. Visweswaran, G. F. Cooper, and G. Clermont. "Outlier detection for patient monitoring and alerting." In: *Journal of Biomedical Informatics* 46.1 (2013), pp. 47–55. ISSN: 1532-0464. DOI: https://doi.org/10.1016/j.jbi.2012.08.004.

[84] D. Heckerman. *Probabilistic similarity networks*. ACM Doctoral dissertation awards. MIT Press, 1991.

[85] D. Heckerman, D. Geiger, and D. M. Chickering. "Learning Bayesian Networks: The Combination of Knowledge and Statistical Data." In: *Machine Learning* 20.3 (1995), pp. 197–243. ISSN: 1573-0565. DOI: 10.1023/A:1022623210503.

[86] M. van der Heijden, M. Velikova, and P. J. F. Lucas. "Learning Bayesian networks for clinical time series analysis." In: *Journal of Biomedical Informatics* 48 (2014), pp. 94–105.

[87] F. Herrera, C. J. Carmona, P. González, and M. J. del Jesus. "An overview on subgroup discovery: foundations and applications." In: *Knowledge and Information Systems* 29.3 (2011), pp. 495–525. ISSN: 0219-3116. DOI: 10.1007/s10115-010-0356-2.

[88] A. Holzinger, C. Biemann, C. S. Pattichis, and D. B. Kell. "What do we need to build explainable AI systems for the medical domain?" In: *CoRR* abs/1712.09923 (2017). arXiv: 1712.09923.

[89] C. L. Hooper and D Bakish. "An examination of the sensitivity of the six-item Hamilton Rating Scale for Depression in a sample of patients suffering from major depressive disorder." In: *Journal of Psychiatry and Neuroscience*. 178th ser. 25.2 (2000), pp. 178–84.

[90] B. Hosenfeld, E. H. Bos, K. J. Wardenaar, H. J. Conradi, H. L. J. van der Maas, I. Visser, and P. de Jonge. "Major depressive disorder as a nonlinear dynamic system: bimodality in the frequency distribution of depressive symptoms over time." In: *BMC Psychiatry* 15.1 (2015), p. 222. ISSN: 1471-244X. DOI: 10.1186/s12888-015-0596-5.

[91] Z. Huang, W. Dong, F. Wang, and H Duan. "Medical Inpatient Journey Modeling and Clustering: A Bayesian Hidden Markov Model Based Approach." In: *AMIA Annual Symposium Proceedings*. 2015, pp. 649–658.

[92] Z. Huang, W. Dong, P. Bath, L. Ji, and H. Duan. "On mining latent treatment patterns from electronic medical records." In: *Data Mining and Knowledge Discovery* 29.4 (2015), pp. 914–949. ISSN: 1573-756X. DOI: 10.1007/s10618-014-0381-y.

[93] Z. Huang, Z. Ge, W. Dong, K. He, and H. Duan. "Probabilistic modeling personalized treatment pathways using electronic health records." In: *Journal of Biomedical Informatics* 86 (2018), pp. 33–48. ISSN: 1532-0464. DOI: https://doi.org/10.1016/j.jbi.2018.08.004.

[94] A. L. Huntley, R. Johnson, S. Purdy, J. M. Valderas, and C. Salisbury. "Measures of Multimorbidity and Morbidity Burden for Use in Primary Care and Community Settings: A Systematic Review and Guide." In: *The Annals of Family Medicine* 10.2 (2012), pp. 134–141. DOI: 10.1370/afm.1363.

[95] M. Hyvärinen, J. Tuomilehto, T. Laatikainen, S. Söderberg, M. Eliasson, P. Nilsson, and Q. Qiao. "The impact of diabetes on coronary heart disease differs from that on ischaemic stroke with regard to the gender." In: *Cardiovascular Diabetology* 8.1 (2009), p. 17. ISSN: 1475-2840. DOI: 10.1186/1475-2840-8-17.

[96] W. W. IsHak, W. Bonifay, K. Collison, M. Reid, H. Youssef, T. Parisi, R. M. Cohen, and L. Cai. "The recovery index: A novel approach to measuring recovery and predicting remission in major depressive disorder." In: *Journal of Affective Disorders* 208 (2017), pp. 369–374. ISSN: 0165-0327. DOI: https://doi.org/10.1016/j.jad.2016.08.081.

[97] F. V. Jensen and T. D. Nielsen. *Bayesian Networks and Decision Graphs*. 2nd. Springer Publishing Company, Incorporated, 2007. ISBN: 9780387682815.

[98] C. C. Jorge, M. Atzmueller, B. M. Heravi, J. L. Gibson, C. R. de Sá, and R. J. F. Rossetti. "Mining Exceptional Social Behaviour." In: *Progress in Artificial Intelligence*. Ed. by P. Moura Oliveira, P. Novais, and L. P. Reis. Cham: Springer International Publishing, 2019, pp. 460–472. ISBN: 978-3-030-30244-3.

[99] B.-H. Juang and L. Rabiner. "Mixture autoregressive hidden Markov models for speech signals." In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 33.6 (1985), pp. 1404–1413. ISSN: 0096-3518.

[100] M. Kaytoue, M. Plantevit, A. Zimmermann, A. Bendimerad, and C. Robardet. "Exceptional contextual subgraph mining." In: *Machine Learning* 106.8 (2017), pp. 1171–1211. ISSN: 1573-0565. DOI: 10.1007/s10994-016-5598-0.

[101] D. Kim, J. Burge, T. Lane, G. Pearlson, K. Kiehl, and V. Calhoun. "Hybrid ICA-Bayesian network approach reveals distinct effective connectivity differences in schizophrenia." In: *NeuroImage* 42.4 (2008), pp. 1560–1568.

[102] S. Kirshner, P. Smyth, and A. Robertson. "Conditional Chow-Liu Tree Structures for Modeling Discrete-Valued Vector Time Series." In: *Proceedings of the 20th Conference in Uncertainty in Artificial Intelligence, Banff, Canada*. 2004, pp. 317–314.

[103] M. Klein and G. Modena. "Estimating Mental States of a Depressed Person with Bayesian Networks." English. In: *Contemporary Challenges and Solutions in Applied Artificial Intelligence*. Ed. by M. Ali, T. Bosse, K. V. Hindriks, M. Hoogendoorn, C. M. Jonker, and J. Treur. Vol. 489. Studies in Computational Intelligence. Springer International Publishing, 2013, pp. 163–168.

[104] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, 2009. ISBN: 978-0-262-01319-2.

[105] M. Längkvist, L. Karlsson, and A. Loutfi. "Sleep Stage Classification Using Unsupervised Feature Learning." In: *Adv. Artif. Neu. Sys.* 2012 (Jan. 2012), 5:5–5:5. ISSN: 1687-7594. DOI: 10.1155/2012/107046.

[106] M. Lappenschaar, A. Hommersom, P. J. F. Lucas, J. Lagro, and S. Visscher. "Multilevel Bayesian networks for the analysis of hierarchical health care data." In: *Artificial Intelligence in Medicine* 57.3 (2013), pp. 171–183. ISSN: 0933-3657. DOI: 10.1016/j.artmed.2012.12.007.

[107] M. Lappenschaar, A. Hommersom, P. J. F. Lucas, J. Lagro, S. Visscher, J. C. Korevaar, and F. G. Schellevis. "Multilevel temporal Bayesian networks can model longitudinal change in multimorbidity." In: *Journal of Clinical Epidemiology* 66.12 (2013), pp. 1405–1416. ISSN: 0895-4356. DOI: 10.1016/j.jclinepi.2013.06.018.

[108] P. Larrañaga, M. Poza, Y. Yurramendi, R. H. Murga, and C. M. H. Kuijpers. "Structure Learning of Bayesian Networks by Genetic Algorithms: A Performance Analysis of Control Parameters." In: *IEEE Trans. Pattern Anal. Mach. Intell.* 18.9 (Sept. 1996), pp. 912–926.

[109] S. Lèbre, J. Becq, F. Devaux, M. Stumpf, and G. Lelandais. "Statistical inference of the time-varying structure of gene-regulation networks." English. In: *BMC Systems Biology* 4.1, 130 (2010), pp. 1–16.

[110] D. Leman, A. Feelders, and A. Knobbe. "Exceptional Model Mining." In: *Machine Learning and Knowledge Discovery in Databases*. Ed. by W. Daelemans, B. Goethals, and K. Morik. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 1–16. ISBN: 978-3-540-87481-2.

[111] F. Lemmerich, M. Atzmueller, and F Puppe. "Fast exhaustive subgroup discovery with numerical target concepts." In: *Data Min Knowl Disc* 30.3 (2016), pp. 711–762. DOI: 10.1007/s10618-015-0436-8.

[112] F. Lemmerich, M. Becker, P. Singer, D. Helic, A. Hotho, and M. Strohmaier. "Mining Subgroups with Exceptional Transition Behavior." In: *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '16. San Francisco, California, USA: ACM, 2016, pp. 965–974. ISBN: 978-1-4503-4232-2. DOI: 10.1145/2939672.2939752.

[113] J. Lijffijt, E. Spyropoulou, B. Kang, and T. De Bie. "P-N-RMiner: a generic framework for mining interesting structured relational patterns." In: *International Journal of Data Science and Analytics* 1.1 (2016), pp. 61–76. ISSN: 2364-4168. DOI: 10.1007/s41060-016-0004-3.

[114] S. Liu, X. Wang, M. Liu, and J. Zhu. "Towards better analysis of machine learning models: A visual analytics perspective." In: *Visual Informatics* 1.1 (2017), pp. 48–56. ISSN: 2468-502X. DOI: https://doi.org/10.1016/j.visinf.2017.01.006.

[115] P. J. F. Lucas. "Certainty-factor-like structures in Bayesian belief networks." In: *Knowledge-based Systems* (2001), pp. 99–119.

[116] B. Malone, K. Kangas, M. Järvisalo, M. Koivisto, and P. Myllymäki. "Predicting the Hardness of Learning Bayesian Networks." In: *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*. AAAI'14. Québec City, Québec, Canada: AAAI Press, 2014, pp. 2460–2466.

[117] E. Manning and M. Gagnon. "The complex patient: A concept clarification." In: *Nursing & Health Sciences* 19.1 (2017), pp. 13–21. DOI: 10.1111/nhs.12320.

[118] D. Margaritis. "Learning Bayesian network model structure from data." PhD thesis. Carnegie Mellon University, 2003.

[119] K. Markov, J. Dang, and S. Nakamura. "Integration of articulatory and spectrum features based on the hybrid HMM/BN modeling framework." In: *Speech Communication* 48.2 (2006), pp. 161–175.

[120] A. Marshall, C. Vasilakis, and E. El-Darzi. "Length of Stay-Based Patient Flow Models: Recent Developments and Future Directions." In: *Health Care Management Science* 8.3 (2005), pp. 213–220. ISSN: 1572-9389. DOI: 10.1007/s10729-005-2012-z.

[121] J. Meier, A. Dietz, A. Boehm, and T. Neumuth. "Predicting treatment process steps from events." In: *Journal of Biomedical Informatics* 53 (2015), pp. 308–319. ISSN: 1532-0464. DOI: https://doi.org/10.1016/j.jbi.2014.12.003.

[122] G. Melançon and F. Philippe. "Generating connected acyclic digraphs uniformly at random." In: *Information Processing Letters* 90.4 (2004), pp. 209–213.

[123] A. Motzek and R. Möller. "Indirect Causes in Dynamic Bayesian Networks Revisited." In: *Proc. of the 24th Intl. Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina*. 2015, pp. 703–709.

[124] K. P. Murphy. "Dynamic Bayesian Networks: Representation, Inference and Learning." PhD thesis. UC Berkeley, Computer Science Division, July 2002.

[125] O. T. Mytton, N. G. Forouhi, P. Scarborough, M. Lentjes, R. Luben, M. Rayner, K. T. Khaw, N. J. Wareham, and P. Monsivais. "Association between intake of less-healthy foods defined by the United Kingdom's nutrient profile model and cardiovascular disease: A population-based cohort study." In: *PLOS Medicine* 15.1 (Jan. 2018), pp. 1–17. DOI: 10.1371/journal.pmed.1002484.

[126] A. Najjar, D. Reinharz, C. Girouard, and C. Gagné. "A two-step approach for mining patient treatment pathways in administrative healthcare databases." In: *Artificial Intelligence in Medicine* 87 (2018), pp. 34–48. ISSN: 0933-3657. DOI: https://doi.org/10.1016/j.artmed.2018.03.004.

[127] *NIVEL Primary Care Database*. https://www.nivel.nl/en/dossier/nivel-primary-care-database. Accessed: 30/04/2018.

[128] P. K. Novak, N. Lavrač, and G. I. Webb. "Supervised Descriptive Rule Discovery: A Unifying Survey of Contrast Set, Emerging Pattern and Subgroup Mining." In: *J. Mach. Learn. Res.* 10 (June 2009), pp. 377–403. ISSN: 1532-4435.

[129] G. O'Donovan, I. Lee, M. Hamer, and E. Stamatakis. "Association of "weekend warrior" and other leisure time physical activity patterns with risks for all-cause, cardiovascular disease, and cancer mortality." In: *JAMA Internal Medicine* 177.3 (2017), pp. 335–342. DOI: 10.1001/jamainternmed.2016.8014.

[130] A. Oniśko and M. J. Druzdzel. "Impact of precision of Bayesian network parameters on accuracy of medical diagnostic systems." In: *Artificial Intelligence in Medicine* 57.3 (2013), pp. 197–206. ISSN: 0933-3657.

[131] A. Oniśko and M. J. Druzdzel. "Impact of Bayesian Network Model Structure on the Accuracy of Medical Diagnostic Systems." In: *Artificial Intelligence and Soft Computing: 13th International Conference, ICAISC 2014, Zakopane, Poland, June 1-5, 2014, Proceedings, Part II*. Ed. by L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L. A. Zadeh, and J. M. Zurada. Cham: Springer International Publishing, 2014, pp. 167–178.

[132] K. Orphanou, A. Stassopoulou, and E. Keravnou. "Temporal abstraction and temporal Bayesian networks in clinical domains: A survey." In: *Artificial Intelligence in Medicine* 60.3 (2014), pp. 133–149.

[133] M. Paoletti, G. Camiciottoli, E. Meoni, F. Bigazzi, L. Cestelli, M. Pistolesi, and C. Marchesi. "Explorative data analysis techniques and unsupervised clustering methods to support clinical assessment of Chronic Obstructive Pulmonary Disease (COPD) phenotypes." In: *Journal of Biomedical Informatics* 42.6 (2009), pp. 1013–1021. ISSN: 1532-0464. DOI: https://doi.org/10.1016/j.jbi.2009.05.008.

[134] M. P. Paulus, M. B. Stein, M. G. Craske, S. Bookheimer, C. T. Taylor, A. N. Simmons, N. Sidhu, K. S. Young, and B. Fan. "Latent variable analysis of positive and negative valence processing focused on symptom and behavioral units of analysis in mood and anxiety disorders." In: *Journal of Affective Disorders* 216 (2017), pp. 17–29. ISSN: 0165-0327. DOI: https://doi.org/10.1016/j.jad.2016.12.046.

[135] S. Pauwels and T. Calders. "An Anomaly Detection Technique for Business Processes based on Extended Dynamic Bayesian Networks." In: *The 34th ACM/SIGAPP Symposium on Applied Computing (SAC '19)*. Limassol, Cyprus: ACM, 2019, pp. 1–8. ISBN: 978-1-4503-5933-7/19/04. DOI: https://doi.org/10.1145/3297280.3297326.

[136] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1988. ISBN: 0-934613-73-7.

[137] J. Pensar, H. Nyman, J. Lintusaari, and J. Corander. "The role of local partial independence in learning of Bayesian networks." In: *International Journal of Approximate Reasoning* 69 (2016), pp. 91–105.

[138] J. Pohle, R. Langrock, F. M. van Beest, and N. M. Schmidt. "Selecting the Number of States in Hidden Markov Models: Pragmatic Solutions Illustrated Using Animal Movement." In: *Journal of Agricultural, Biological and Environmental Statistics* 22.3 (2017), pp. 270–293. ISSN: 1537-2693. DOI: 10.1007/s13253-017-0283-8.

[139] A. Poritz. "Linear predictive hidden Markov models and the speech signal." In: *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '82*. Vol. 7. 1982, pp. 1291–1294.

[140] A. Prados-Torres, B. Poblador-Plou, A. Calderón-Larrañaga, L. A. Gimeno-Feliu, F. González-Rubio, A. Poncel-Falcó, A. Sicras-Mainar, and J. T. Alcalá-Nalvaiz. "Multimorbidity Patterns in Primary Care: Interactions among Chronic Diseases Using Factor Analysis." In: *PLOS ONE* 7.2 (Feb. 2012), pp. 1–12. DOI: 10.1371/journal.pone.0032190.

[141] L. R. Rabiner. "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition." In: *Readings in Speech Recognition*. Ed. by A. Waibel and K.-F. Lee. San Francisco: Morgan Kaufmann, 1990, pp. 267–296. ISBN: 978-1-55860-124-6. DOI: https://doi.org/10.1016/B978-0-08-051584-7.50027-9.

[142] A. Rajkomar et al. "Scalable and accurate deep learning with electronic health records." In: *npj Digital Medicine* 1.1 (2018), p. 18. DOI: 10.1038/s41746-018-0029-1.

[143] C. Riggelsen. "Learning Bayesian Networks from Incomplete Data: An Efficient Method for Generating Approximate Predictive Distributions." In: *Proceedings of the Sixth SIAM International Conference on Data Mining*. Bethesda, MD, USA, 2006, pp. 130–140. DOI: 10.1137/1.9781611972764.12.

[144] A. W. Robertson, S. Kirshner, and P. Smyth. "Downscaling of Daily Rainfall Occurrence over Northeast Brazil Using a Hidden Markov Model." In: *Journal of Climate* 17.22 (2004), pp. 4407–4424.

[145] J. W. Robinson and A. J. Hartemink. "Learning Non-Stationary Dynamic Bayesian Networks." In: *J. Mach. Learn. Res.* 11 (Dec. 2010), pp. 3647–3680. ISSN: 1532-4435.

[146] C. Rose, C. Smaili, and F. Charpillet. "A dynamic Bayesian network for handling uncertainty in a decision support system adapted to the monitoring of patients treated by hemodialysis." In: *Tools with Artificial Intelligence, 2005. ICTAI 05. 17th IEEE International Conference on*. 2005, 5 pp.–598.

[147] A. J. Rothschild. "Challenges in the Treatment of Major Depressive Disorder With Psychotic Features." In: *Schizophrenia Bulletin* 39.4 (2013), pp. 787–796. DOI: 10.1093/schbul/sbt046.

[148] A. Rozinat, M. Veloso, and W. van der Aalst. "Evaluating the Quality of Discovered Process Models." In: *Proc. of Induction of Process Models (ECML PKDD)*. Antwerp, Belgium, 2008, pp. 45–52.

[149] C. R. de Sá, W. Duivesteijn, P. Azevedo, A. M. Jorge, C. Soares, and A. Knobbe. "Discovering a taste for the unusual: exceptional models for preference mining." In: *Machine Learning* 107.11 (2018), pp. 1775–1807. ISSN: 1573-0565. DOI: 10.1007/s10994-018-5743-z.

[150] M. Scanagatta, C. P. de Campos, G. Corani, and M. Zaffalon. "Learning Bayesian Networks with Thousands of Variables." In: *Advances in Neural Information Processing Systems 28*. Ed. by C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett. Curran Associates, Inc., 2015, pp. 1864–1872.

[151] M. Scutari and J. Denis. *Bayesian Networks with Examples in R*. Boca Raton: Chapman and Hall, 2014.

[152] M. Scutari. "Learning Bayesian Networks with the bnlearn R Package." In: *Journal of Statistical Software* 35.3 (2010), pp. 1–22. DOI: 10.18637/jss.v035.i03.

[153] F. L. Seixas, B. Zadrozny, J. Laks, A. Conci, and D. C. M. Saade. "A Bayesian network decision model for supporting the diagnosis of dementia, Alzheimer's disease and mild cognitive impairment." In: *Computers in Biology and Medicine* 51 (2014), pp. 140–158.

[154] H. J. Seltman, S. Mitchell, and R. A. Sweet. "A Bayesian model of psychosis symptom trajectory in Alzheimer's disease." In: *International Journal of Geriatric Psychiatry* 31.2 (2016), pp. 204–210. DOI: 10.1002/gps.4326.

[155] K. Seymore, A. Mccallum, and R. Rosenfeld. "Learning Hidden Markov Model Structure for Information Extraction." In: *AAAI 99 Workshop on Mach. Learning for Inf. Extr.* 1999, pp. 37–42.

[156] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas. "Taking the Human Out of the Loop: A Review of Bayesian Optimization." In: *Proceedings of the IEEE* 104.1 (2016), pp. 148–175. ISSN: 0018-9219. DOI: 10.1109/JPROC.2015.2494218.

[157] Z. Shojaei Estabragh, M. Riahi Kashani, F. Jeddi Moghaddam, S. Sari, Z. Taherifar, S. Moradi Moosavy, and K. Sadeghi Oskooyee. "Bayesian network modeling for diagnosis of social anxiety using some cognitive-behavioral factors." English. In: *Network Modeling Analysis in Health Informatics and Bioinformatics* 2.4 (2013), pp. 257–265.

[158] T. Silander and P. Myllymäki. "A Simple Approach for Finding the Globally Optimal Bayesian Network Structure." In: *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence*. UAI'06. Cambridge, MA, USA: AUAI Press, 2006, pp. 445–452. ISBN: 0-9749039-2-2.

[159] J. Sinnige, J. C. Korevaar, G. P. Westert, P. Spreeuwenberg, F. G. Schellevis, and J. C. Braspenning. "Multimorbidity patterns in a primary care population aged 55 years and over." In: *Family Practice* 32.5 (2015), pp. 505–513.

[160] J. Snoek, H. Larochelle, and R. P. Adams. "Practical Bayesian Optimization of Machine Learning Algorithms." In: *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 2*. NIPS'12. Lake Tahoe, Nevada: Curran Associates Inc., 2012, pp. 2951–2959.

[161] H. Song. "Model-Based Subgroup Discovery." PhD thesis. University of Bristol, Nov. 2017. DOI: 10.13140/RG.2.2.19978.57280.

[162] S. Sorias. "Bayesian networks: Overcoming the Limitations of the Descriptive and Categorical Approaches in Psychiatric Diagnosis. A proposal based on Bayesian networks." In: *Turkish Journal of Psychiatry* (2014), pp. 1–12. ISSN: 1300-2163.

[163] P. Spirtes and C. Glymour. "An Algorithm for Fast Recovery of Sparse Causal Graphs." In: *Social Science Computer Review* 9.1 (1991), pp. 62–72. DOI: 10.1177/089443939100900106.

[164] E. Stamatakis, L. F. M. de Rezende, and J. P. Rey-López. "Sedentary Behaviour and Cardiovascular Disease." In: *Sedentary Behaviour Epidemiology*. Ed. by M. F. Leitzmann, C. Jochem, and D. Schmid. Cham: Springer Intl. Publishing, 2018, pp. 215–243. ISBN: 978-3-319-61552-3.

[165] M. Stanke, O. Schöffmann, B. Morgenstern, and S. Waack. "Gene prediction in eukaryotes with a generalized hidden Markov model that uses hints from external sources." In: *BMC Bioinformatics* 7 (2006), p. 62.

[166] W. Steeman. *BPI Challenge 2013, incidents*. 2013. DOI: 10.4121/uuid:500573e6-accc-4b0c-9576-aa5468b10cee.

[167] B. van Strien. "Exceptional Model Mining of Convolutional Neural Networks." MSc thesis. Technical University of Eindhoven, 2019.

[168] K. Tokuda, Y. Nankaku, T. Toda, H. Zen, J. Yamagishi, and K. Oura. "Speech Synthesis Based on Hidden Markov Models." In: *Proceedings of the IEEE* 101.5 (2013), pp. 1234–1252. ISSN: 0018-9219.

[169] I. Tsamardinos, L. E. Brown, and C. F. Aliferis. "The max-min hill-climbing Bayesian network structure learning algorithm." In: *Machine Learning* 65.1 (2006), pp. 31–78. ISSN: 1573-0565. DOI: 10.1007/s10994-006-6889-7.

[170] A. Tucker and X. Liu. "Learning Dynamic Bayesian Networks from Multivariate Time Series with Changing Dependencies." English. In: *Advances in Intelligent Data Analysis V*. Ed. by M. R. Berthold, H.-J. Lenz, E. Bradley, R. Kruse, and C. Borgelt. Vol. 2810. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2003, pp. 100–110.

[171] S. Visscher, P. J. F. Lucas, I. Flesch, and K. Schurink. "Using Temporal Context-Specific Independence Information in the Exploratory Analysis of Disease Processes." English. In: *Art. Int. in Medicine*. Ed. by R. Bellazzi et al. Vol. 4594. Lecture Notes in Computer Science. Springer Berlin H., 2007, pp. 87–96.

[172] S. D. Vito, M. Piga, L. Martinotto, and G. D. Francia. "CO, NO2 and NOx urban pollution monitoring with on-field calibrated electronic nose by automatic bayesian regularization." In: *Sensors and Actuators B: Chemical* 143.1 (2009), pp. 182–191. ISSN: 0925-4005.

[173] J. Vlasselaer, W. Meert, G. van den Broeck, and L. Raedt. "Exploiting local and repeated structure in Dynamic Bayesian Networks." In: *Artificial Intelligence* 232 (2016), pp. 43–53.

[174] L. Wangikar, S. Dhuwalia, A. Yadav, B. Dikshit, and D. Yadav. *Faster Payments to Farmers: Analysis of the Direct Payments Process of EU's Agricultural Guarantee Fund – Business Process Intelligence Challenge 2018*. 2018.

[175] J. Wijkstra et al. "Treatment of unipolar psychotic depression: a randomized, double-blind study comparing imipramine, venlafaxine, and venlafaxine plus quetiapine." In: *Acta Psychiatrica Scandinavica* 121.3 (2009), pp. 190–200. ISSN: 1600-0447.

[176] J. Wijkstra et al. "Treatment of unipolar psychotic depression: a randomized, double-blind study comparing imipramine, venlafaxine, and venlafaxine plus quetiapine." In: *Acta Psychiatrica Scandinavica* 121.3 (2010), pp. 190–200. DOI: 10.1111/j.1600-0447.2009.01464.x.

[177] J Wijkstra, J Lijmer, H Burger, A Cipriani, J Geddes, and W. Nolen. "Pharmacological treatment for psychotic depression." In: *Cochrane Database of Systematic Reviews* 7 (2015). ISSN: 1465-1858. DOI: 10.1002/14651858.CD004044.pub4.

[178] B. Yet, Z. Perkins, N. Fenton, N. Tai, and W. Marsh. "Not just data: A method for improving prediction with knowledge." In: *Journal of Biomedical Informatics* 48 (2014), pp. 28–37. ISSN: 1532-0464. DOI: https://doi.org/10.1016/j.jbi.2013.10.012.

[179] N. L. Zhang, T. D. Nielsen, and F. V. Jensen. "Latent variable discovery in classification models." In: *Artificial Intelligence in Medicine* 30.3 (2004). Bayesian Networks in Biomedicince and Health-Care, pp. 283–299. ISSN: 0933-3657. DOI: https://doi.org/10.1016/j.artmed.2003.11.004.

[180] Y. Zhang, Q. Lin, J. Wang, and S. Verwer. "Car-following Behavior Model Learning Using Timed Automata." In: *IFAC-PapersOnLine* 50.1 (2017). 20th IFAC World Congress, pp. 2353–2358. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2017.08.423.

[181] A. Zimmermann and L. De Raedt. "Cluster-grouping: from subgroup discovery to clustering." In: *Machine Learning* 77.1 (2009), pp. 125–159. ISSN: 1573-0565. DOI: 10.1007/s10994-009-5121-y.

[182] W. Zucchini. "An Introduction to Model Selection." In: *J. Math. Psychol.* 44.1 (Mar. 2000), pp. 41–61.

SUMMARY

Temporal processes, such as walking, sleeping, eating, and so on, are ubiquitous in daily life as well as in more intricate situations such as medical treatment, seasonal climate variation, events in a workflow, and so on. We are often interested in understanding how aspects of objects of study evolve, such as signs and symptoms of disorders of patients. On the one hand, we need *expressive* enough models for capturing complex behavior. On the other hand, such models should provide *parsimonious* descriptions of processes if one wishes to gain insight. This balance is not trivial, as by increasing expressivity one often arrives at more complicated models, which might make them less interpretable. It is often the case that suitable descriptions of processes also need that uncertainty be explicitly recognized.

In this thesis, we aim to increase model expressivity inspired by complex and real-life problems, while retaining model interpretability. To this end, we describe three new different viewpoints on processes based on probabilistic graphical models.

We first provide a new process viewpoint based on *latent states*, which can be seen as abstract representations of the observable data. Latent states can help interpretation as they act as a dimensionality reduction tool.

In Chapter 3, we introduce asymmetric hidden Markov models for capturing local structure in the space of observable variables. This is done by associating each latent state to a Bayesian network. Asymmetric hidden Markov models often lead to better model fit and increased insight into the domain, while reducing the need for selecting an *a priori* model architecture. Simulated and real-world datasets are used for empirical evaluation.

In Chapter 4, we propose a semi-automatic framework for understanding disease dynamics based on the dynamics of latent states within hidden Markov models. We apply the framework to psychotic depression treatments, where latent states act as patient groups and are shown to reveal predictive symptoms to patient prognosis.

In Chapter 5, we learn hidden Markov models from health-care event data. A case study based on atherosclerosis events is used. The size of such datasets, in contrast with a small number of events, makes the same event be associated to multiple hidden states, a notion we call clustering of hidden states. We show that events in a cluster associate to patients with different disease severity.

The second viewpoint on processes is based on the identification of process *change-points* or *regime change*. The challenge lies in how to extend models that are time invariant (such as dynamic Bayesian networks) for capturing regime change in a parsimonious way, which can be suitable when the available dataset is small.

In Chapter 6, we propose partitioned dynamic Bayesian networks for representing models for which the time homogeneity assumption is not suitable. Partitioned dynamic Bayesian networks are a collection of dynamic Bayesian networks for which cut-off points are built heuristically. The resulting models are evaluated in a wide set of experiments.

The last process viewpoint tries to discover subsets of temporal data associated to models that deviate substantially from the model obtained from the whole dataset. This can be seen as identifying significant *subprocesses*.

In Chapter 7, we introduce dynamic Bayesian networks for representing exceptional temporal models of data subgroups. This provides a general representation for subprocesses within the context of subgroup discovery and exceptional model mining. We evaluate the proposed approach by means of simulated and event data on farmer financial support applications.

SAMENVATTING

Temporele processen beschrijven gebeurtenissen in het dagelijks leven, zoals lopen, slapen en eten, etc., maar ook meer gecompliceerde situaties zoals medische behandelingen, de jaargetijden, gebeurtenissen in een workflow, etc. We zijn gewoonlijk geïnteresseerd om te begrijpen hoe bepaalde aspecten van objecten zich ontwikkelen, zoals de ziektesymptomen van een patiënt. Aan de ene kant hebben we *expressieve* modellen nodig om complex gedrag vast te leggen. Aan de andere kant moeten dergelijke modellen *compacte* beschrijvingen van processen opleveren om inzicht te verwerven. Het vinden van de juiste deze balans tussen deze twee kenmerken van modellen is niet triviaal, want door het verhogen van de expressiviteit komt men vaak tot complexere modellen, die wellicht minder interpreteerbaar zijn. Vaak is het ook nog nodig dat geschikte beschrijvingen van processen expliciet rekening houden met onzekerheid.

In dit proefschrift willen we de expressiviteit van het model vergroten, geïnspireerd door de complexiteit van reële problemen, met behoud van de interpreteerbaarheid van het model. Daartoe beschrijven we drie nieuwe verschillende gezichtspunten op processen op basis van probabilistisch grafische modellen.

We geven eerst een nieuw procesperspectief op basis van *latente toestanden*, die kunnen worden gezien als abstracte representaties van de waarneembare data. Latente toestanden kunnen helpen bij de interpretatie, omdat ze fungeren als een instrument voor *dimensionaliteitvermindering*.

In hoofdstuk 3 introduceren we asymmetrische hidden Markov modellen voor het vastleggen van de lokale structuur tussen de waarneembare variabelen. Dit wordt gedaan door elke latente toestand te associëren met een Bayesiaans netwerk. Asymmetrische hidden Markov-modellen leiden vaak tot een betere kwaliteit van modellen en meer inzicht in het domein, terwijl de noodzaak van het kiezen van een a priori modelarchitectuur wordt verminderd. Een empirische evaluatie werd uitgevoerd met behulp van gesimuleerde en echte datasets.

In hoofdstuk 4 stellen we een semi-automatisch raamwerk voor om ziekteprocessen te begrijpen op basis van de dynamiek van latente toestanden binnen de hidden Markov modellen. We hebben het raamwerk toegepast op gegevens die verkregen zijn bij de behandeling van patiënten met psychotische depressie, waarbij latente toestanden als patiëntgroepen fungeren, die symptomen voorspellen als onderdeel van de prognose van de patiënt.

In hoofdstuk 5 leren we hidden Markov modellen uit gegevens van gebeurtenissen in de gezondheidszorg. Er wordt gebruik gemaakt van een casus op basis van gebeurtenissen die te maken hebben met aderverkalking. De grootte van dergelijke datasets, in vergelijking met het kleine aantal mogelijke gebeurtenissen in de datasets, maakt dat dezelfde gebeurtenis wordt geassocieerd met meerdere latente toestanden, een begrip dat we clustering van latente toestanden noe-

men. We laten zien dat gebeurtenissen in een cluster geassocieerd worden met patiënten met een verschillende ernst van de ziekte.

Het tweede gezichtspunt op processen is gebaseerd op de identificatie van procesveranderingspunten of regimeverandering. De uitdaging ligt in het uitbreiden van modellen die tijdsinvariant zijn (zoals dynamische Bayesiaanse netwerken) voor het vastleggen van regimeverandering op een eenvoudige manier, die geschikt kan zijn wanneer de beschikbare dataset klein is.

In hoofdstuk 6 stellen we gepartitioneerde dynamische Bayesiaanse netwerken voor om modellen te kunnen bouwen waarvoor de tijdshomogeniteitsaanname niet geschikt is. Gepartitioneerde dynamische Bayesiaanse netwerken zijn een verzameling van dynamische Bayesiaanse netwerken waarbij afkappunten heuristisch worden geïdentificeerd. Deze modellen werden geëvalueerd in een brede verzameling experimenten.

In het laatste procesperspectief wordt getracht deelverzamelingen van tijdsgegevens te ontdekken die samenhangen met modellen die substantieel afwijken van het model dat uit de hele dataset wordt verkregen. Dit kan worden gezien als het identificeren van belangrijke subprocessen.

In hoofdstuk 7 introduceren we dynamische Bayesiaanse netwerken voor het representeren van uitzonderlijke temporele modellen uit deelverzamelingen van de data. Dit biedt een algemene representatie voor subprocessen binnen de context van de ontdekking van subgroepen en *exceptional model mining*. We evalueren de voorgestelde aanpak door middel van gesimuleerde data en een casus rond subsidieaanvragen in de agrarische sector.

ACKNOWLEDGMENTS

A PhD is a long and challenging journey. I would like to acknowledge the people who helped me make this possible. But being a PhD student also teaches you a lot about yourself, such as your limitations and strengths, and this part seems to be very individual.

Many say that a PhD is to a great extent about learning how to do independent research, which although seemingly simple, is actually a deep statement about methodology. One of the difficulties of the daily grind of a PhD trajectory is to not forget this when one is immersed in the development of a new algorithm or implementation. Somehow this helps to better handle failures, something I would have never imagined to have such an importance in research.

I learned many valuable lessons from my supervisors, Peter Lucas and Arjen Hommersom. First and foremost, thank you Peter and Arjen for taking me as a PhD student. You helped me develop a much broader view on research and helped me become a better researcher. Peter showed me the importance of having a global perspective about my research, which sometimes made me step back so that I could see farther and overcome deadlocks. Arjen has been willing to discuss many technical details of my research, and patient enough to hear my sometimes confusing ideas and contribute to them. During the meetings with my supervisors, I was very often faced with a critical view in a level that was new to me, something not always easy but important to strengthen ideas in research. We also had interesting discussions about science, culture, cuisine and many other things.

Some chapters of this thesis resulted from a collaboration with Joost Janzing from UMC Radboud. In this collaboration I was exposed to new viewpoints on research, particularly on medicine and medical AI, which was a rich experience. Thank you, Joost, for willing to collaborate and for our insightful discussions.

As a visiting researcher, I spent 6 months at CINTESIS, in Porto, Portugal. Although short, it was an interesting experience where I learned more about research on medical AI. One chapter of this thesis was done in collaboration with Pedro Rodrigues from CINTESIS. I thank Pedro and CINTESIS for hosting me during that time.

I would like to thank the co-authors with whom I collaborated at different moments: Alexis, Martijn, Mariana and Sicco. I also thank Océ Technologies for our collaboration on machine learning for cyber-physical systems.

I am grateful for the time I spent at the software science section of iCIS, Radboud University Nijmegen as a PhD candidate. Ingrid, thank you very much for assisting me before and after landing in Nijmegen with all sorts of administrative stuff. I enjoyed our conversations about Dutch culture a lot. I also thank all the software science staff, and Harco Kuppens for helping with

technical support. I also thank LIACS at Leiden University for hosting me at the final stage of my PhD studies.

I had the chance to have great colleagues at iCIS with whom I worked or shared an office. I thank my office mate Giso for the many insightful discussions we had on our research on probabilistic graphical models, on computer science, as well as for introducing me to the world of good coffee. I had many interesting discussions with Paul, Manxia and Steffen about research and doing research. I also thank Alexis, David, Joshua, Jurriën, Marina, Martijn, Markus and Tim for the pleasant conversations during coffee breaks, *borrelen*, etc.

Some of our earliest friends in Nijmegen were Annika and Sophia. We spent such a great time with them, thank you! I also thank Simone Meeuwsen for the friendship, for having helped us a great deal when moving houses in Nijmegen and for willing to hear my attempts of speaking some Dutch. Also thanks to Boris and Bowon for the interesting conversations about living abroad and for the pleasant meetings.

Agradeço aos meus pais, Celina e Marcos, e meu irmão Matheus, pelo amor e incentivo durante todo o doutorado, em especial ao meu pai pelo incentivo em buscar a carreira acadêmica. Por fim mas não menos importante, agradeço à minha esposa Ilka pelo amor e pelo incentivo em todos os momentos; juntos realizamos e construimos sonhos ao longo dessa jornada nos Países Baixos.

CURRICULUM VITAE

Marcos Luiz de Paula Bueno

1984	Born in Catalão, Brazil
2003–2007	B.Sc. in Computer Science Federal University of Goiás, Brazil
2008–2010	M.Sc. in Computer Science Federal University of Uberlândia, Brazil
2011–2014	Assistant professor Department of Computer Science Federal University of Uberlândia, Brazil
2014–2020	PhD student iCIS, Radboud University Nijmegen, The Netherlands LIACS, Leiden University, The Netherlands
2017	Visiting researcher (6 months) CINTESIS, University of Porto, Portugal
2018–	Assistant professor Department of Computer Science Federal University of Uberlândia, Brazil

SIKS DISSERTATIONS

2011 01 Botond Cseke (RUN), Variational Algorithms for Bayesian Inference in Latent Gaussian Models
02 Nick Tinnemeier (UU), Organizing Agent Organizations. Syntax and Operational Semantics of an Organization-Oriented Programming Language
03 Jan Martijn van der Werf (TUE), Compositional Design and Verification of Component-Based Information Systems
04 Hado van Hasselt (UU), Insights in Reinforcement Learning; Formal analysis and empirical evaluation of temporal-difference
05 Bas van der Raadt (VU), Enterprise Architecture Coming of Age - Increasing the Performance of an Emerging Discipline.
06 Yiwen Wang (TUE), Semantically-Enhanced Recommendations in Cultural Heritage
07 Yujia Cao (UT), Multimodal Information Presentation for High Load Human Computer Interaction
08 Nieske Vergunst (UU), BDI-based Generation of Robust Task-Oriented Dialogues
09 Tim de Jong (OU), Contextualised Mobile Media for Learning
10 Bart Bogaert (UvT), Cloud Content Contention
11 Dhaval Vyas (UT), Designing for Awareness: An Experience-focused HCI Perspective
12 Carmen Bratosin (TUE), Grid Architecture for Distributed Process Mining
13 Xiaoyu Mao (UvT), Airport under Control. Multiagent Scheduling for Airport Ground Handling
14 Milan Lovric (EUR), Behavioral Finance and Agent-Based Artificial Markets
15 Marijn Koolen (UvA), The Meaning of Structure: the Value of Link Evidence for Information Retrieval
16 Maarten Schadd (UM), Selective Search in Games of Different Complexity
17 Jiyin He (UVA), Exploring Topic Structure: Coherence, Diversity and Relatedness
18 Mark Ponsen (UM), Strategic Decision-Making in complex games
19 Ellen Rusman (OU), The Mind's Eye on Personal Profiles
20 Qing Gu (VU), Guiding service-oriented software engineering - A view-based approach
21 Linda Terlouw (TUD), Modularization and Specification of Service-Oriented Systems
22 Junte Zhang (UVA), System Evaluation of Archival Description and Access
23 Wouter Weerkamp (UVA), Finding People and their Utterances in Social Media
24 Herwin van Welbergen (UT), Behavior Generation for Interpersonal Coordination with Virtual Humans On Specifying, Scheduling and Realizing Multimodal Virtual Human Behavior
25 Syed Waqar ul Qounain Jaffry (VU), Analysis and Validation of Models for Trust Dynamics
26 Matthijs Aart Pontier (VU), Virtual Agents for Human Communication - Emotion Regulation and Involvement-Distance Trade-Offs in Embodied Conversational Agents and Robots
27 Aniel Bhulai (VU), Dynamic website optimization through autonomous management of design patterns
28 Rianne Kaptein (UVA), Effective Focused Retrieval by Exploiting Query Context and Document Structure
29 Faisal Kamiran (TUE), Discrimination-aware Classification
30 Egon van den Broek (UT), Affective Signal Processing (ASP): Unraveling the mystery of emotions
31 Ludo Waltman (EUR), Computational and Game-Theoretic Approaches for Modeling Bounded Rationality
32 Nees-Jan van Eck (EUR), Methodological Advances in Bibliometric Mapping of Science
33 Tom van der Weide (UU), Arguing to Motivate Decisions
34 Paolo Turrini (UU), Strategic Reasoning in Interdependence: Logical and Game-theoretical Investigations

35 Maaike Harbers (UU), Explaining Agent Behavior in Virtual Training
36 Erik van der Spek (UU), Experiments in serious game design: a cognitive approach
37 Adriana Burlutiu (RUN), Machine Learning for Pairwise Data, Applications for Preference Learning and Supervised Network Inference
38 Nyree Lemmens (UM), Bee-inspired Distributed Optimization
39 Joost Westra (UU), Organizing Adaptation using Agents in Serious Games
40 Viktor Clerc (VU), Architectural Knowledge Management in Global Software Development
41 Luan Ibraimi (UT), Cryptographically Enforced Distributed Data Access Control
42 Michal Sindlar (UU), Explaining Behavior through Mental State Attribution
43 Henk van der Schuur (UU), Process Improvement through Software Operation Knowledge
44 Boris Reuderink (UT), Robust Brain-Computer Interfaces
45 Herman Stehouwer (UvT), Statistical Language Models for Alternative Sequence Selection
46 Beibei Hu (TUD), Towards Contextualized Information Delivery: A Rule-based Architecture for the Domain of Mobile Police Work
47 Azizi Bin Ab Aziz (VU), Exploring Computational Models for Intelligent Support of Persons with Depression
48 Mark Ter Maat (UT), Response Selection and Turn-taking for a Sensitive Artificial Listening Agent
49 Andreea Niculescu (UT), Conversational interfaces for task-oriented spoken dialogues: design aspects influencing interaction quality

---

2012 01 Terry Kakeeto (UvT), Relationship Marketing for SMEs in Uganda
02 Muhammad Umair (VU), Adaptivity, emotion, and Rationality in Human and Ambient Agent Models
03 Adam Vanya (VU), Supporting Architecture Evolution by Mining Software Repositories
04 Jurriaan Souer (UU), Development of Content Management System-based Web Applications
05 Marijn Plomp (UU), Maturing Interorganisational Information Systems
06 Wolfgang Reinhardt (OU), Awareness Support for Knowledge Workers in Research Networks
07 Rianne van Lambalgen (VU), When the Going Gets Tough: Exploring Agent-based Models of Human Performance under Demanding Conditions
08 Gerben de Vries (UVA), Kernel Methods for Vessel Trajectories
09 Ricardo Neisse (UT), Trust and Privacy Management Support for Context-Aware Service Platforms
10 David Smits (TUE), Towards a Generic Distributed Adaptive Hypermedia Environment
11 J.C.B. Rantham Prabhakara (TUE), Process Mining in the Large: Preprocessing, Discovery, and Diagnostics
12 Kees van der Sluijs (TUE), Model Driven Design and Data Integration in Semantic Web Information Systems
13 Suleman Shahid (UvT), Fun and Face: Exploring non-verbal expressions of emotion during playful interactions
14 Evgeny Knutov (TUE), Generic Adaptation Framework for Unifying Adaptive Web-based Systems
15 Natalie van der Wal (VU), Social Agents. Agent-Based Modelling of Integrated Internal and Social Dynamics of Cognitive and Affective Processes.
16 Fiemke Both (VU), Helping people by understanding them - Ambient Agents supporting task execution and depression treatment
17 Amal Elgammal (UvT), Towards a Comprehensive Framework for Business Process Compliance
18 Eltjo Poort (VU), Improving Solution Architecting Practices
19 Helen Schonenberg (TUE), What's Next? Operational Support for Business Process Execution
20 Ali Bahramisharif (RUN), Covert Visual Spatial Attention, a Robust Paradigm for Brain-Computer Interfacing
21 Roberto Cornacchia (TUD), Querying Sparse Matrices for Information Retrieval
22 Thijs Vis (UvT), Intelligence, politie en veiligheidsdienst: verenigbare grootheden?
23 Christian Muehl (UT), Toward Affective Brain-Computer Interfaces: Exploring the Neurophysiology of Affect during Human Media Interaction
24 Laurens van der Werff (UT), Evaluation of Noisy Transcripts for Spoken Document Retrieval

25 Silja Eckartz (UT), Managing the Business Case Development in Inter-Organizational IT Projects: A Methodology and its Application
26 Emile de Maat (UVA), Making Sense of Legal Text
27 Hayrettin Gurkok (UT), Mind the Sheep! User Experience Evaluation & Brain-Computer Interface Games
28 Nancy Pascall (UvT), Engendering Technology Empowering Women
29 Almer Tigelaar (UT), Peer-to-Peer Information Retrieval
30 Alina Pommeranz (TUD), Designing Human-Centered Systems for Reflective Decision Making
31 Emily Bagarukayo (RUN), A Learning by Construction Approach for Higher Order Cognitive Skills Improvement, Building Capacity and Infrastructure
32 Wietske Visser (TUD), Qualitative multi-criteria preference representation and reasoning
33 Rory Sie (OUN), Coalitions in Cooperation Networks (COCOON)
34 Pavol Jancura (RUN), Evolutionary analysis in PPI networks and applications
35 Evert Haasdijk (VU), Never Too Old To Learn – On-line Evolution of Controllers in Swarm- and Modular Robotics
36 Denis Ssebugwawo (RUN), Analysis and Evaluation of Collaborative Modeling Processes
37 Agnes Nakakawa (RUN), A Collaboration Process for Enterprise Architecture Creation
38 Selmar Smit (VU), Parameter Tuning and Scientific Testing in Evolutionary Algorithms
39 Hassan Fatemi (UT), Risk-aware design of value and coordination networks
40 Agus Gunawan (UvT), Information Access for SMEs in Indonesia
41 Sebastian Kelle (OU), Game Design Patterns for Learning
42 Dominique Verpoorten (OU), Reflection Amplifiers in self-regulated Learning
43 Withdrawn
44 Anna Tordai (VU), On Combining Alignment Techniques
45 Benedikt Kratz (UvT), A Model and Language for Business-aware Transactions
46 Simon Carter (UVA), Exploration and Exploitation of Multilingual Data for Statistical Machine Translation
47 Manos Tsagkias (UVA), Mining Social Media: Tracking Content and Predicting Behavior
48 Jorn Bakker (TUE), Handling Abrupt Changes in Evolving Time-series Data
49 Michael Kaisers (UM), Learning against Learning - Evolutionary dynamics of reinforcement learning algorithms in strategic interactions
50 Steven van Kervel (TUD), Ontologogy driven Enterprise Information Systems Engineering
51 Jeroen de Jong (TUD), Heuristics in Dynamic Sceduling; a practical framework with a case study in elevator dispatching

---

2013 01 Viorel Milea (EUR), News Analytics for Financial Decision Support
02 Erietta Liarou (CWI), MonetDB/DataCell: Leveraging the Column-store Database Technology for Efficient and Scalable Stream Processing
03 Szymon Klarman (VU), Reasoning with Contexts in Description Logics
04 Chetan Yadati (TUD), Coordinating autonomous planning and scheduling
05 Dulce Pumareja (UT), Groupware Requirements Evolutions Patterns
06 Romulo Goncalves (CWI), The Data Cyclotron: Juggling Data and Queries for a Data Warehouse Audience
07 Giel van Lankveld (UvT), Quantifying Individual Player Differences
08 Robbert-Jan Merk (VU), Making enemies: cognitive modeling for opponent agents in fighter pilot simulators
09 Fabio Gori (RUN), Metagenomic Data Analysis: Computational Methods and Applications
10 Jeewanie Jayasinghe Arachchige (UvT), A Unified Modeling Framework for Service Design.
11 Evangelos Pournaras (TUD), Multi-level Reconfigurable Self-organization in Overlay Services
12 Marian Razavian (VU), Knowledge-driven Migration to Services
13 Mohammad Safiri (UT), Service Tailoring: User-centric creation of integrated IT-based homecare services to support independent living of elderly
14 Jafar Tanha (UVA), Ensemble Approaches to Semi-Supervised Learning Learning
15 Daniel Hennes (UM), Multiagent Learning - Dynamic Games and Applications
16 Eric Kok (UU), Exploring the practical benefits of argumentation in multi-agent deliberation
17 Koen Kok (VU), The PowerMatcher: Smart Coordination for the Smart Electricity Grid
18 Jeroen Janssens (UvT), Outlier Selection and One-Class Classification

19 Renze Steenhuizen (TUD), Coordinated Multi-Agent Planning and Scheduling
20 Katja Hofmann (UvA), Fast and Reliable Online Learning to Rank for Information Retrieval
21 Sander Wubben (UvT), Text-to-text generation by monolingual machine translation
22 Tom Claassen (RUN), Causal Discovery and Logic
23 Patricio de Alencar Silva (UvT), Value Activity Monitoring
24 Haitham Bou Ammar (UM), Automated Transfer in Reinforcement Learning
25 Agnieszka Anna Latoszek-Berendsen (UM), Intention-based Decision Support. A new way of representing and implementing clinical guidelines in a Decision Support System
26 Alireza Zarghami (UT), Architectural Support for Dynamic Homecare Service Provisioning
27 Mohammad Huq (UT), Inference-based Framework Managing Data Provenance
28 Frans van der Sluis (UT), When Complexity becomes Interesting: An Inquiry into the Information eXperience
29 Iwan de Kok (UT), Listening Heads
30 Joyce Nakatumba (TUE), Resource-Aware Business Process Management: Analysis and Support
31 Dinh Khoa Nguyen (UvT), Blueprint Model and Language for Engineering Cloud Applications
32 Kamakshi Rajagopal (OUN), Networking For Learning; The role of Networking in a Lifelong Learner's Professional Development
33 Qi Gao (TUD), User Modeling and Personalization in the Microblogging Sphere
34 Kien Tjin-Kam-Jet (UT), Distributed Deep Web Search
35 Abdallah El Ali (UvA), Minimal Mobile Human Computer Interaction
36 Than Lam Hoang (TUe), Pattern Mining in Data Streams
37 Dirk Börner (OUN), Ambient Learning Displays
38 Eelco den Heijer (VU), Autonomous Evolutionary Art
39 Joop de Jong (TUD), A Method for Enterprise Ontology based Design of Enterprise Information Systems
40 Pim Nijssen (UM), Monte-Carlo Tree Search for Multi-Player Games
41 Jochem Liem (UVA), Supporting the Conceptual Modelling of Dynamic Systems: A Knowledge Engineering Perspective on Qualitative Reasoning
42 Léon Planken (TUD), Algorithms for Simple Temporal Reasoning
43 Marc Bron (UVA), Exploration and Contextualization through Interaction and Concepts

---

2014 01 Nicola Barile (UU), Studies in Learning Monotone Models from Data
02 Fiona Tuliyano (RUN), Combining System Dynamics with a Domain Modeling Method
03 Sergio Raul Duarte Torres (UT), Information Retrieval for Children: Search Behavior and Solutions
04 Hanna Jochmann-Mannak (UT), Websites for children: search strategies and interface design - Three studies on children's search performance and evaluation
05 Jurriaan van Reijsen (UU), Knowledge Perspectives on Advancing Dynamic Capability
06 Damian Tamburri (VU), Supporting Networked Software Development
07 Arya Adriansyah (TUE), Aligning Observed and Modeled Behavior
08 Samur Araujo (TUD), Data Integration over Distributed and Heterogeneous Data Endpoints
09 Philip Jackson (UvT), Toward Human-Level Artificial Intelligence: Representation and Computation of Meaning in Natural Language
10 Ivan Salvador Razo Zapata (VU), Service Value Networks
11 Janneke van der Zwaan (TUD), An Empathic Virtual Buddy for Social Support
12 Willem van Willigen (VU), Look Ma, No Hands: Aspects of Autonomous Vehicle Control
13 Arlette van Wissen (VU), Agent-Based Support for Behavior Change: Models and Applications in Health and Safety Domains
14 Yangyang Shi (TUD), Language Models With Meta-information
15 Natalya Mogles (VU), Agent-Based Analysis and Support of Human Functioning in Complex Socio-Technical Systems: Applications in Safety and Healthcare
16 Krystyna Milian (VU), Supporting trial recruitment and design by automatically interpreting eligibility criteria
17 Kathrin Dentler (VU), Computing healthcare quality indicators automatically: Secondary Use of Patient Data and Semantic Interoperability
18 Mattijs Ghijsen (UVA), Methods and Models for the Design and Study of Dynamic Agent Organizations

19 Vinicius Ramos (TUE), Adaptive Hypermedia Courses: Qualitative and Quantitative Evaluation and Tool Support
20 Mena Habib (UT), Named Entity Extraction and Disambiguation for Informal Text: The Missing Link
21 Kassidy Clark (TUD), Negotiation and Monitoring in Open Environments
22 Marieke Peeters (UU), Personalized Educational Games - Developing agent-supported scenario-based training
23 Eleftherios Sidirourgos (UvA/CWI), Space Efficient Indexes for the Big Data Era
24 Davide Ceolin (VU), Trusting Semi-structured Web Data
25 Martijn Lappenschaar (RUN), New network models for the analysis of disease interaction
26 Tim Baarslag (TUD), What to Bid and When to Stop
27 Rui Jorge Almeida (EUR), Conditional Density Models Integrating Fuzzy and Probabilistic Representations of Uncertainty
28 Anna Chmielowiec (VU), Decentralized k-Clique Matching
29 Jaap Kabbedijk (UU), Variability in Multi-Tenant Enterprise Software
30 Peter de Cock (UvT), Anticipating Criminal Behaviour
31 Leo van Moergestel (UU), Agent Technology in Agile Multiparallel Manufacturing and Product Support
32 Naser Ayat (UvA), On Entity Resolution in Probabilistic Data
33 Tesfa Tegegne (RUN), Service Discovery in eHealth
34 Christina Manteli (VU), The Effect of Governance in Global Software Development: Analyzing Transactive Memory Systems.
35 Joost van Ooijen (UU), Cognitive Agents in Virtual Worlds: A Middleware Design Approach
36 Joos Buijs (TUE), Flexible Evolutionary Algorithms for Mining Structured Process Models
37 Maral Dadvar (UT), Experts and Machines United Against Cyberbullying
38 Danny Plass-Oude Bos (UT), Making brain-computer interfaces better: improving usability through post-processing.
39 Jasmina Maric (UvT), Web Communities, Immigration, and Social Capital
40 Walter Omona (RUN), A Framework for Knowledge Management Using ICT in Higher Education
41 Frederic Hogenboom (EUR), Automated Detection of Financial Events in News Text
42 Carsten Eijckhof (CWI/TUD), Contextual Multidimensional Relevance Models
43 Kevin Vlaanderen (UU), Supporting Process Improvement using Method Increments
44 Paulien Meesters (UvT), Intelligent Blauw. Met als ondertitel: Intelligence-gestuurde politiezorg in gebiedsgebonden eenheden.
45 Birgit Schmitz (OUN), Mobile Games for Learning: A Pattern-Based Approach
46 Ke Tao (TUD), Social Web Data Analytics: Relevance, Redundancy, Diversity
47 Shangsong Liang (UVA), Fusion and Diversification in Information Retrieval

---

2015 01 Niels Netten (UvA), Machine Learning for Relevance of Information in Crisis Response
02 Faiza Bukhsh (UvT), Smart auditing: Innovative Compliance Checking in Customs Controls
03 Twan van Laarhoven (RUN), Machine learning for network data
04 Howard Spoelstra (OUN), Collaborations in Open Learning Environments
05 Christoph Bösch (UT), Cryptographically Enforced Search Pattern Hiding
06 Farideh Heidari (TUD), Business Process Quality Computation - Computing Non-Functional Requirements to Improve Business Processes
07 Maria-Hendrike Peetz (UvA), Time-Aware Online Reputation Analysis
08 Jie Jiang (TUD), Organizational Compliance: An agent-based model for designing and evaluating organizational interactions
09 Randy Klaassen (UT), HCI Perspectives on Behavior Change Support Systems
10 Henry Hermans (OUN), OpenU: design of an integrated system to support lifelong learning
11 Yongming Luo (TUE), Designing algorithms for big graph datasets: A study of computing bisimulation and joins
12 Julie M. Birkholz (VU), Modi Operandi of Social Network Dynamics: The Effect of Context on Scientific Collaboration Networks
13 Giuseppe Procaccianti (VU), Energy-Efficient Software
14 Bart van Straalen (UT), A cognitive approach to modeling bad news conversations
15 Klaas Andries de Graaf (VU), Ontology-based Software Architecture Documentation

16 Changyun Wei (UT), Cognitive Coordination for Cooperative Multi-Robot Teamwork
17 André van Cleeff (UT), Physical and Digital Security Mechanisms: Properties, Combinations and Trade-offs
18 Holger Pirk (CWI), Waste Not, Want Not! - Managing Relational Data in Asymmetric Memories
19 Bernardo Tabuenca (OUN), Ubiquitous Technology for Lifelong Learners
20 Lois Vanhée (UU), Using Culture and Values to Support Flexible Coordination
21 Sibren Fetter (OUN), Using Peer-Support to Expand and Stabilize Online Learning
22 Zhemin Zhu (UT), Co-occurrence Rate Networks
23 Luit Gazendam (VU), Cataloguer Support in Cultural Heritage
24 Richard Berendsen (UVA), Finding People, Papers, and Posts: Vertical Search Algorithms and Evaluation
25 Steven Woudenberg (UU), Bayesian Tools for Early Disease Detection
26 Alexander Hogenboom (EUR), Sentiment Analysis of Text Guided by Semantics and Structure
27 Sándor Héman (CWI), Updating compressed colomn stores
28 Janet Bagorogoza (TiU), Knowledge Management and High Performance; The Uganda Financial Institutions Model for HPO
29 Hendrik Baier (UM), Monte-Carlo Tree Search Enhancements for One-Player and Two-Player Domains
30 Kiavash Bahreini (OU), Real-time Multimodal Emotion Recognition in E-Learning
31 Yakup Koç (TUD), On the robustness of Power Grids
32 Jerome Gard (UL), Corporate Venture Management in SMEs
33 Frederik Schadd (TUD), Ontology Mapping with Auxiliary Resources
34 Victor de Graaf (UT), Gesocial Recommender Systems
35 Jungxao Xu (TUD), Affective Body Language of Humanoid Robots: Perception and Effects in Human Robot Interaction

---

2016 01 Syed Saiden Abbas (RUN), Recognition of Shapes by Humans and Machines
02 Michiel Christiaan Meulendijk (UU), Optimizing medication reviews through decision support: prescribing a better pill to swallow
03 Maya Sappelli (RUN), Knowledge Work in Context: User Centered Knowledge Worker Support
04 Laurens Rietveld (VU), Publishing and Consuming Linked Data
05 Evgeny Sherkhonov (UVA), Expanded Acyclic Queries: Containment and an Application in Explaining Missing Answers
06 Michel Wilson (TUD), Robust scheduling in an uncertain environment
07 Jeroen de Man (VU), Measuring and modeling negative emotions for virtual training
08 Matje van de Camp (TiU), A Link to the Past: Constructing Historical Social Networks from Unstructured Data
09 Archana Nottamkandath (VU), Trusting Crowdsourced Information on Cultural Artefacts
10 George Karafotias (VUA), Parameter Control for Evolutionary Algorithms
11 Anne Schuth (UVA), Search Engines that Learn from Their Users
12 Max Knobbout (UU), Logics for Modelling and Verifying Normative Multi-Agent Systems
13 Nana Baah Gyan (VU), The Web, Speech Technologies and Rural Development in West Africa - An ICT4D Approach
14 Ravi Khadka (UU), Revisiting Legacy Software System Modernization
15 Steffen Michels (RUN), Hybrid Probabilistic Logics - Theoretical Aspects, Algorithms and Experiments
16 Guangliang Li (UVA), Socially Intelligent Autonomous Agents that Learn from Human Reward
17 Berend Weel (VU), Towards Embodied Evolution of Robot Organisms
18 Albert Meroño Peñuela (VU), Refining Statistical Data on the Web
19 Julia Efremova (Tu/e), Mining Social Structures from Genealogical Data
20 Daan Odijk (UVA), Context & Semantics in News & Web Search
21 Alejandro Moreno Célleri (UT), From Traditional to Interactive Playspaces: Automatic Analysis of Player Behavior in the Interactive Tag Playground
22 Grace Lewis (VU), Software Architecture Strategies for Cyber-Foraging Systems
23 Fei Cai (UVA), Query Auto Completion in Information Retrieval

24 Brend Wanders (UT), Repurposing and Probabilistic Integration of Data; An Iterative and data model independent approach
25 Julia Kiseleva (TU/e), Using Contextual Information to Understand Searching and Browsing Behavior
26 Dilhan Thilakarathne (VU), In or Out of Control: Exploring Computational Models to Study the Role of Human Awareness and Control in Behavioural Choices, with Applications in Aviation and Energy Management Domains
27 Wen Li (TUD), Understanding Geo-spatial Information on Social Media
28 Mingxin Zhang (TUD), Large-scale Agent-based Social Simulation - A study on epidemic prediction and control
29 Nicolas Höning (TUD), Peak reduction in decentralised electricity systems - Markets and prices for flexible planning
30 Ruud Mattheij (UvT), The Eyes Have It
31 Mohammad Khelghati (UT), Deep web content monitoring
32 Eelco Vriezekolk (UT), Assessing Telecommunication Service Availability Risks for Crisis Organisations
33 Peter Bloem (UVA), Single Sample Statistics, exercises in learning from just one example
34 Dennis Schunselaar (TUE), Configurable Process Trees: Elicitation, Analysis, and Enactment
35 Zhaochun Ren (UVA), Monitoring Social Media: Summarization, Classification and Recommendation
36 Daphne Karreman (UT), Beyond R2D2: The design of nonverbal interaction behavior optimized for robot-specific morphologies
37 Giovanni Sileno (UvA), Aligning Law and Action - a conceptual and computational inquiry
38 Andrea Minuto (UT), Materials that Matter - Smart Materials meet Art & Interaction Design
39 Merijn Bruijnes (UT), Believable Suspect Agents; Response and Interpersonal Style Selection for an Artificial Suspect
40 Christian Detweiler (TUD), Accounting for Values in Design
41 Thomas King (TUD), Governing Governance: A Formal Framework for Analysing Institutional Design and Enactment Governance
42 Spyros Martzoukos (UVA), Combinatorial and Compositional Aspects of Bilingual Aligned Corpora
43 Saskia Koldijk (RUN), Context-Aware Support for Stress Self-Management: From Theory to Practice
44 Thibault Sellam (UVA), Automatic Assistants for Database Exploration
45 Bram van de Laar (UT), Experiencing Brain-Computer Interface Control
46 Jorge Gallego Perez (UT), Robots to Make you Happy
47 Christina Weber (UL), Real-time foresight - Preparedness for dynamic innovation networks
48 Tanja Buttler (TUD), Collecting Lessons Learned
49 Gleb Polevoy (TUD), Participation and Interaction in Projects. A Game-Theoretic Analysis
50 Yan Wang (UVT), The Bridge of Dreams: Towards a Method for Operational Performance Alignment in IT-enabled Service Supply Chains

---

2017 01 Jan-Jaap Oerlemans (UL), Investigating Cybercrime
02 Sjoerd Timmer (UU), Designing and Understanding Forensic Bayesian Networks using Argumentation
03 Daniël Harold Telgen (UU), Grid Manufacturing; A Cyber-Physical Approach with Autonomous Products and Reconfigurable Manufacturing Machines
04 Mrunal Gawade (CWI), Multi-core Parallelism in a Column-store
05 Mahdieh Shadi (UVA), Collaboration Behavior
06 Damir Vandic (EUR), Intelligent Information Systems for Web Product Search
07 Roel Bertens (UU), Insight in Information: from Abstract to Anomaly
08 Rob Konijn (VU) , Detecting Interesting Differences:Data Mining in Health Insurance Data using Outlier Detection and Subgroup Discovery
09 Dong Nguyen (UT), Text as Social and Cultural Data: A Computational Perspective on Variation in Text
10 Robby van Delden (UT), (Steering) Interactive Play Behavior
11 Florian Kunneman (RUN), Modelling patterns of time and emotion in Twitter #anticipointment
12 Sander Leemans (TUE), Robust Process Mining with Guarantees

13 Gijs Huisman (UT), Social Touch Technology - Extending the reach of social touch through haptic technology
14 Shoshannah Tekofsky (UvT), You Are Who You Play You Are: Modelling Player Traits from Video Game Behavior
15 Peter Berck (RUN), Memory-Based Text Correction
16 Aleksandr Chuklin (UVA), Understanding and Modeling Users of Modern Search Engines
17 Daniel Dimov (UL), Crowdsourced Online Dispute Resolution
18 Ridho Reinanda (UVA), Entity Associations for Search
19 Jeroen Vuurens (UT), Proximity of Terms, Texts and Semantic Vectors in Information Retrieval
20 Mohammadbashir Sedighi (TUD), Fostering Engagement in Knowledge Sharing: The Role of Perceived Benefits, Costs and Visibility
21 Jeroen Linssen (UT), Meta Matters in Interactive Storytelling and Serious Gaming (A Play on Worlds)
22 Sara Magliacane (VU), Logics for causal inference under uncertainty
23 David Graus (UVA), Entities of Interest — Discovery in Digital Traces
24 Chang Wang (TUD), Use of Affordances for Efficient Robot Learning
25 Veruska Zamborlini (VU), Knowledge Representation for Clinical Guidelines, with applications to Multimorbidity Analysis and Literature Search
26 Merel Jung (UT), Socially intelligent robots that understand and respond to human touch
27 Michiel Joosse (UT), Investigating Positioning and Gaze Behaviors of Social Robots: People's Preferences, Perceptions and Behaviors
28 John Klein (VU), Architecture Practices for Complex Contexts
29 Adel Alhuraibi (UvT), From IT-BusinessStrategic Alignment to Performance: A Moderated Mediation Model of Social Innovation, and Enterprise Governance of IT"
30 Wilma Latuny (UvT), The Power of Facial Expressions
31 Ben Ruijl (UL), Advances in computational methods for QFT calculations
32 Thaer Samar (RUN), Access to and Retrievability of Content in Web Archives
33 Brigit van Loggem (OU), Towards a Design Rationale for Software Documentation: A Model of Computer-Mediated Activity
34 Maren Scheffel (OU), The Evaluation Framework for Learning Analytics
35 Martine de Vos (VU), Interpreting natural science spreadsheets
36 Yuanhao Guo (UL), Shape Analysis for Phenotype Characterisation from High-throughput Imaging
37 Alejandro Montes Garcia (TUE), WiBAF: A Within Browser Adaptation Framework that Enables Control over Privacy
38 Alex Kayal (TUD), Normative Social Applications
39 Sara Ahmadi (RUN), Exploiting properties of the human auditory system and compressive sensing methods to increase noise robustness in ASR
40 Altaf Hussain Abro (VUA), Steer your Mind: Computational Exploration of Human Control in Relation to Emotions, Desires and Social Support For applications in human-aware support systems
41 Adnan Manzoor (VUA), Minding a Healthy Lifestyle: An Exploration of Mental Processes and a Smart Environment to Provide Support for a Healthy Lifestyle
42 Elena Sokolova (RUN), Causal discovery from mixed and missing data with applications on ADHD datasets
43 Maaike de Boer (RUN), Semantic Mapping in Video Retrieval
44 Garm Lucassen (UU), Understanding User Stories - Computational Linguistics in Agile Requirements Engineering
45 Bas Testerink (UU), Decentralized Runtime Norm Enforcement
46 Jan Schneider (OU), Sensor-based Learning Support
47 Jie Yang (TUD), Crowd Knowledge Creation Acceleration
48 Angel Suarez (OU), Collaborative inquiry-based learning

2018 01 Han van der Aa (VUA), Comparing and Aligning Process Representations
02 Felix Mannhardt (TUE), Multi-perspective Process Mining
03 Steven Bosems (UT), Causal Models For Well-Being: Knowledge Modeling, Model-Driven Development of Context-Aware Applications, and Behavior Prediction

04 Jordan Janeiro (TUD), Flexible Coordination Support for Diagnosis Teams in Data-Centric Engineering Tasks
05 Hugo Huurdeman (UVA), Supporting the Complex Dynamics of the Information Seeking Process
06 Dan Ionita (UT), Model-Driven Information Security Risk Assessment of Socio-Technical Systems
07 Jieting Luo (UU), A formal account of opportunism in multi-agent systems
08 Rick Smetsers (RUN), Advances in Model Learning for Software Systems
09 Xu Xie (TUD), Data Assimilation in Discrete Event Simulations
10 Julienka Mollee (VUA), Moving forward: supporting physical activity behavior change through intelligent technology
11 Mahdi Sargolzaei (UVA), Enabling Framework for Service-oriented Collaborative Networks
12 Xixi Lu (TUE), Using behavioral context in process mining
13 Seyed Amin Tabatabaei (VUA), Computing a Sustainable Future
14 Bart Joosten (UVT), Detecting Social Signals with Spatiotemporal Gabor Filters
15 Naser Davarzani (UM), Biomarker discovery in heart failure
16 Jaebok Kim (UT), Automatic recognition of engagement and emotion in a group of children
17 Jianpeng Zhang (TUE), On Graph Sample Clustering
18 Henriette Nakad (UL), De Notaris en Private Rechtspraak
19 Minh Duc Pham (VUA), Emergent relational schemas for RDF
20 Manxia Liu (RUN), Time and Bayesian Networks
21 Aad Slootmaker (OUN), EMERGO: a generic platform for authoring and playing scenario-based serious games
22 Eric Fernandes de Mello Araujo (VUA), Contagious: Modeling the Spread of Behaviours, Perceptions and Emotions in Social Networks
23 Kim Schouten (EUR), Semantics-driven Aspect-Based Sentiment Analysis
24 Jered Vroon (UT), Responsive Social Positioning Behaviour for Semi-Autonomous Telepresence Robots
25 Riste Gligorov (VUA), Serious Games in Audio-Visual Collections
26 Roelof Anne Jelle de Vries (UT),Theory-Based and Tailor-Made: Motivational Messages for Behavior Change Technology
27 Maikel Leemans (TUE), Hierarchical Process Mining for Scalable Software Analysis
28 Christian Willemse (UT), Social Touch Technologies: How they feel and how they make you feel
29 Yu Gu (UVT), Emotion Recognition from Mandarin Speech
30 Wouter Beek, The "K" in "semantic web" stands for "knowledge": scaling semantics to the web

2019 01 Rob van Eijk (UL),Web privacy measurement in real-time bidding systems. A graph-based approach to RTB system classification
02 Emmanuelle Beauxis Aussalet (CWI, UU), Statistics and Visualizations for Assessing Class Size Uncertainty
03 Eduardo Gonzalez Lopez de Murillas (TUE), Process Mining on Databases: Extracting Event Data from Real Life Data Sources
04 Ridho Rahmadi (RUN), Finding stable causal structures from clinical data
05 Sebastiaan van Zelst (TUE), Process Mining with Streaming Data
06 Chris Dijkshoorn (VU), Nichesourcing for Improving Access to Linked Cultural Heritage Datasets
07 Soude Fazeli (TUD), Recommender Systems in Social Learning Platforms
08 Frits de Nijs (TUD), Resource-constrained Multi-agent Markov Decision Processes
09 Fahimeh Alizadeh Moghaddam (UVA), Self-adaptation for energy efficiency in software systems
10 Qing Chuan Ye (EUR), Multi-objective Optimization Methods for Allocation and Prediction
11 Yue Zhao (TUD), Learning Analytics Technology to Understand Learner Behavioral Engagement in MOOCs
12 Jacqueline Heinerman (VU), Better Together
13 Guanliang Chen (TUD), MOOC Analytics: Learner Modeling and Content Generation
14 Daniel Davis (TUD), Large-Scale Learning Analytics: Modeling Learner Behavior & Improving Learning Outcomes in Massive Open Online Courses

15 Erwin Walraven (TUD), Planning under Uncertainty in Constrained and Partially Observable Environments
16 Guangming Li (TUE), Process Mining based on Object-Centric Behavioral Constraint (OCBC) Models
17 Ali Hurriyetoglu (RUN),Extracting actionable information from microtexts
18 Gerard Wagenaar (UU), Artefacts in Agile Team Communication
19 Vincent Koeman (TUD), Tools for Developing Cognitive Agents
20 Chide Groenouwe (UU), Fostering technically augmented human collective intelligence
21 Cong Liu (TUE), Software Data Analytics: Architectural Model Discovery and Design Pattern Detection
22 Martin van den Berg (VU),Improving IT Decisions with Enterprise Architecture
23 Qin Liu (TUD), Intelligent Control Systems: Learning, Interpreting, Verification
24 Anca Dumitrache (VU), Truth in Disagreement - Crowdsourcing Labeled Data for Natural Language Processing
25 Emiel van Miltenburg (VU), Pragmatic factors in (automatic) image description
26 Prince Singh (UT), An Integration Platform for Synchromodal Transport
27 Alessandra Antonaci (OUN), The Gamification Design Process applied to (Massive) Open Online Courses
28 Esther Kuindersma (UL), Cleared for take-off: Game-based learning to prepare airline pilots for critical situations
29 Daniel Formolo (VU), Using virtual agents for simulation and training of social skills in safety-critical circumstances
30 Vahid Yazdanpanah (UT), Multiagent Industrial Symbiosis Systems
31 Milan Jelisavcic (VU), Alive and Kicking: Baby Steps in Robotics
32 Chiara Sironi (UM), Monte-Carlo Tree Search for Artificial General Intelligence in Games
33 Anil Yaman (TUE), Evolution of Biologically Inspired Learning in Artificial Neural Networks
34 Negar Ahmadi (TUE), EEG Microstate and Functional Brain Network Features for Classification of Epilepsy and PNES
35 Lisa Facey-Shaw (OUN), Gamification with digital badges in learning programming
36 Kevin Ackermans (OUN), Designing Video-Enhanced Rubrics to Master Complex Skills
37 Jian Fang (TUD), Database Acceleration on FPGAs
38 Ákos Kádár (TiU), Learning visually grounded and multilingual representations

2020 01 Armon Toubman (UL), Calculated Moves: Generating Air Combat Behaviour
02 Marcos Luiz de Paula Bueno (UL), Unraveling Temporal Processes using Probabilistic Graphical Models