APPROXIMATION METHODS FOR EFFICIENT LEARNING OF BAYESIAN NETWORKS

SIKS Dissertation Series No. 2006-16

The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

Universiteit Utrecht

Printed by Febodruk B.V., The Netherlands.

ISBN-10: 90-393-4289-X
ISBN-13: 978-90-393-4289-3

Approximation Methods for Efficient Learning of Bayesian Networks

Benaderingsmethoden voor het efficiënt leren van Bayesiaanse netwerken

(met een samenvatting in het Nederlands)

Approximationsmetoder til efficient læring af Bayesianske netværk

(med et resumé på dansk)

Proefschrift

ter verkrijging van de graad van doctor aan de Universiteit Utrecht op gezag van de rector magnificus, prof.dr. W.H. Gispen, ingevolge het besluit van het college voor promoties in het openbaar te verdedigen op maandag 23 oktober 2006 des middags te 14.30 uur

door

Carsten Riggelsen

geboren op 28 september 1975 te Haderslev, Denemarken.

Promotor: Prof.dr. A.P.J.M. Siebes
Co-promotor: Dr. A.J. Feelders

Contents

Foreword

This dissertation is the result of 4 years at Utrecht University as a Ph.D-student at the Department of Information and Computing Sciences. The work presented in this thesis is mainly based on the research published in various papers during that time. It is not a bundle of accepted articles. In order to provide a coherent treatment of matters, I combined the main message of each paper with unpublished work and things that were not discussed or treated in those articles. If read from beginning to end, I hope that my efforts have been worthwhile, and that the reader is able to understand the overall message.

My gratitude goes to my supervisor and co-promotor, Ad Feelders. I am thankful to my promotor, Arno Siebes and to Jeroen De Knijf, Edwin de Jong and all the other colleagues in the Large Distributed Databases group, and at the computer science department. Also I would also like to thank the members of the reading committee, Linda van der Gaag, Richard D. Gill, Finn V. Jensen, Bert Kappen and Pedro Larrañaga.

Carsten Riggelsen
Deventer, August 6, 2006

Chapter 1

INTRODUCTION

1. Learning from data

Several approaches to learning from data have emerged through time, many of them based on very different theories and philosophies. Data mining, machine learning, knowledge discovery, etc. all share the fundamental desire to extract in a principled fashion some *essentials* from data (e.g., local patterns, nuggets, (ir)regularities, etc.) or even learn about the underlying *data generating process*.

Which approach to learning is the most appropriate is open to discussion, and may depend on pragmatic considerations or more profound beliefs. Anyhow, it is almost certain that computer science will play a central role, directly or indirectly, in the learning task. Computer science has a major impact on all fields of scientific research. It pushes the boundaries of "what's possible" in most areas. In many ways, advances in computer science have altered the way we perceive scientific research today.

We consider the task of learning from data as a challenge that is inherently statistical in nature. Like most branches of science, statistics is built upon well-founded principles that stipulates lines of valid reasoning. However, in learning from data, the statistical approach is the natural choice because it provides methods and tools that have been designed specifically for the purpose of analysing (observational) data. Although our departure point to learning is a statistical one, there is an important pragmatic aspect to this; we do not want to constrain ourselves to a pure statistical approach when the statistical theory leads

to computationally intractable implementations. In that case we resort to methods that are perhaps less well-founded or principled in nature, yet are tractable from a computational point of view. Fortunately, with the recent advances in computer science, statistical analysis has proven itself in practice as well.

2. The role of computation

Improvements in computational power means that *Monte Carlo* (MC) simulation has become an viable approach to many statistical problems, that could not be tackled easily before. In the so-called *Bayesian approach*, statistical inference is based on entire distributions on various unknown quantities. The distributions usually have a form that makes it difficult to formulate them as a closed expression. By way of *Markov chain Monte Carlo* (MCMC) simulation, empirical evidence can be produced from these distributions. These empirical samples approximate the posterior distributions, and inference is now based on these samples rather than the exact distributions. MCMC methods are sound and valid from a statistical point of view, but it is only by the help of computers that it has become feasible to actually implement the techniques. This in turn has meant that the Bayesian paradigm, treating "everything" we are uncertain about as a random variable, has received quite a lot of attention recently, and to a large extent is just as feasible as the classical statistical approach. Bayesian statistics is usually more informative compared to the classical approach, as entire distributions convey more information than a few summary statistics.

3. Learning from incomplete data

Most methods for performing statistical data analysis or learning require complete data in order to work or produce valid results. Unfortunately real-life data(bases) are rarely complete. For statistical analysis of incomplete data the standard tools and algorithms developed for complete data often don't suffice anymore. Incomplete data is structured in a way that makes principled data analysis very difficult and sometimes even intractable. Valid statistical analysis of incomplete data often leads to high computational complexity compared to a complete data scenario. Learning from incomplete data is a non-trivial extension of existing methods developed for learning from complete data.

Also in this context, MCMC simulation techniques provide a feasible means for learning within the Bayesian paradigm. However, the added computational burden of the learning task with incomplete data means

that for large amounts of data that even MCMC techniques may be too expensive. In that case the only feasible thing to do is to slightly relax a few assumption about interaction effects in the data, apply heuristics and resort to summary statistics. An unfortunate side effect of this is that we have to trade in the Bayesian approach to data analysis. However, for large amounts of data, posterior distributions are close to Normal distributions anyway, and therefore summary statistics often suffice to describe the distributions.

4. Bayesian network learning

Learning from data ranges between extracting *essentials* from the data, to the more fundamental and very challenging task of learning the underlying *data generating process*. The former only captures certain aspects of the data, whereas the latter captures the very "thing" that gave rise to the data. Obviously, learning the data generating process yields the strongest result in the sense that it reveals "everything" there is to learn.

From a statistical point of view, the assumption is that we can capture the underlying data generating process in terms of a probability distribution. This gives rise to the question which statistical formalism is expressive and flexible enough to be able to capture a broad range of data generating processes. The *Bayesian network* (BN) formalism is one possibility. Informally speaking BNs capture the interaction effects that hold between random variables in terms of a graph. Put in another way, BNs are so-called directed graphical models which is a class of statistical models defined by a collection of conditional independences between variables. This graph offers an appealing way of structuring an otherwise confusing number of equations expressing the (in)dependences between variables.

BNs occupy a prominent position in decision support environments where they are used for diagnostic and prediction purposes. In terms of interpretability, the directed graphical structure (model) of a BN is attractive, because explicit insight is gained into relationships between variables. Learning the underlying data generating process in terms of a BN means that explicit insight is gained into the "workings" of the data generating process.

5. Research objectives

The title of the thesis "Approximation Methods for Efficient Learning of Bayesian Networks" summarises what this thesis is about: We

develop and investigate efficient Monte Carlo simulation methods in order to realise a Bayesian approach to approximate learning of Bayesian networks from both complete and incomplete data. For large amounts of incomplete data when Monte Carlo methods are inefficient, the approximations mentioned in section 3 are implemented, such that learning remains feasible, albeit non-Bayesian.

6. Thesis outline

This thesis is divided into six chapters. The topics treated in each chapter are as follows:

Chapter 2 In the next chapter the notation and some basic concepts about probabilities, graph theory and conditional independence are introduced. We present the Bayesian network formalism, and we illustrate how to do parameter learning taking a Bayesian statistical approach.

Chapter 3 This chapter is about Bayesian network learning from data. We continue where we left off in chapter 2, and focus on the Bayesian approach to model and parameter learning. In that context the so-called marginal likelihood is also introduced which plays a crucial role when learning from both complete and incomplete data.

Chapter 4 Here we treat Monte Carlo simulation techniques, importance sampling and Markov chain Monte Carlo (MCMC) methods. These techniques are used in chapter 5. We also develop an efficient MCMC algorithm called MB-MCMC for sampling Bayesian network models from complete data.

Chapter 5 In this chapter we introduce the concept of incomplete data, and we illustrate the difficulties in connection with Bayesian network learning. We go on and treat principled methods for learning, and discuss iterative methods for model and parameter learning. Various Bayesian approaches to learning based on MCMC simulation are also introduced. We combine importance sampling and MCMC, and develop an efficient algorithm for improved model learning from incomplete data called $e\text{MC}^4$. Finally we treat less principled, yet very fast methods for learning, and a new algorithm called MBP is developed.

Chapter 6 We draw conclusions and summarise the main contributions of this thesis.

Chapter 2

PRELIMINARIES

In this chapter the notation is presented and the basic concepts related to the Bayesian network formalism are treated. Towards the end of the chapter, we introduce the Bayesian statistical approach to learning. Obviously there is much more to say about these issues than what is mentioned here. Thorough introductions and in-depth treatment can be found elsewhere in the literature such as Cowell et al., 1999; Jensen, 2001; Whittaker, 1990.

1. Random variables and conditional independence

Essential to the Bayesian network formalism is the notion of *conditional independence* between *random variables*. In this section we focus on the statistical/probabilistic properties of these variables and the algebraic laws of the (in)dependences between variables; see for instance Feller, 1970 for more details.

Throughout we use upper case letters to denote random variables, X, Y, Z. The corresponding lower case letters denote values of the variables, x, y, z. Bold letters denote vectors, $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}$ for variables and $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$ for values. The term *vector* is used somewhat loosely, and in some places the term *set* would be more appropriate. Coordinate projection is sometimes used to extract the required sub-vector of $\boldsymbol{X}$, e.g, $\boldsymbol{X}_{\{2,5\}} = \boldsymbol{X}_{25}$ denotes the vector (X_2, X_5). Any deviation from these conventions will be clear from the context or will be explicitly mentioned when needed.

A random variable X has a state-space Ω_X consisting of the possible values x. The state-space of a vector of random variables $\boldsymbol{X}$ is the Cartesian product of the individual state-spaces of the variables X_i in $\boldsymbol{X}$, i.e., $\Omega_{\boldsymbol{X}} = \prod_i \times \Omega_{X_i}$.

In the following, for continuous variables, summations should be replaced by integrals. A *joint* probability distribution $\Pr(\boldsymbol{X})$ over $\boldsymbol{X}$ is a mapping $\Omega_{\boldsymbol{X}} \mapsto [0;1]$ with $\sum_{\boldsymbol{x}} \Pr(\boldsymbol{x}) = 1$. The *marginal* distribution over $\boldsymbol{Y}$ where $\boldsymbol{X} = \boldsymbol{Y} \cup \boldsymbol{Z}$ is $\Pr(\boldsymbol{Y}) = \sum_{\boldsymbol{Z}} \Pr(\boldsymbol{z}, \boldsymbol{Y})$, i.e., the marginal distribution is obtained by summing out all variables except $\boldsymbol{Y}$. We say that $\boldsymbol{Y}$ is *independent* of $\boldsymbol{Z}$ if $\Pr(\boldsymbol{Y}, \boldsymbol{Z}) = \Pr(\boldsymbol{Y}) \cdot \Pr(\boldsymbol{Z})$.

The *conditional* probability distribution $\Pr(\boldsymbol{Z}|\boldsymbol{y})$ is defined as $\frac{\Pr(\boldsymbol{Z},\boldsymbol{y})}{\Pr(\boldsymbol{y})}$ for $\Pr(\boldsymbol{y}) > 0$, i.e., re-normalise over $\boldsymbol{Z}$ for fixed $\boldsymbol{Y}$.

Everything is now in place to introduce the concept of *conditional independence*. Conditional independence between random variables captures a notion of irrelevance between variables that comes about when the values of some other variables are given.

Formally we say that $\boldsymbol{Y}$ is *conditionally independent* of $\boldsymbol{Z}$ *given* $\boldsymbol{S}$ if (not surprisingly) $\Pr(\boldsymbol{Z}, \boldsymbol{Y}|\boldsymbol{S}) = \Pr(\boldsymbol{Z}|\boldsymbol{S}) \cdot \Pr(\boldsymbol{Y}|\boldsymbol{S})$. For $\Pr(\boldsymbol{Y}|\boldsymbol{S}) > 0$ we may instead write this as $\Pr(\boldsymbol{Z}|\boldsymbol{Y}, \boldsymbol{S}) = \Pr(\boldsymbol{Z}|\boldsymbol{S})$, i.e., once $\boldsymbol{S}$ is given, $\boldsymbol{Z}$ becomes independent of $\boldsymbol{Y}$.

The interpretation of the latter equation is conveniently captured by the operator $\perp\!\!\!\perp$; the independence statement is then written as $\boldsymbol{Z} \perp\!\!\!\perp \boldsymbol{Y}|\boldsymbol{S}$. Given a set of conditional independence statements, other conditional independence statements are entailed, and they can be derived using the so-called *(semi-)graphoid* axioms provided by Pearl, 1988. There are however many entailed statements, and we therefore introduce a different algebraic structure that captures many of these statements "at a glance"; this structure takes the form of a graph.

2. Graph theory

A graph is a pair $(\boldsymbol{X}, \boldsymbol{E})$, where $\boldsymbol{X} = \{X_1, \ldots, X_p\}$ is a set of p vertices and $\boldsymbol{E}$ is a subset of the ordered pair $\boldsymbol{X} \times \boldsymbol{X}$. If $(X_i, X_j) \in \boldsymbol{E}$ and $(X_j, X_i) \in \boldsymbol{E}$ then vertex X_i and X_j are connected by an *undirected edge*, and we write this as $X_i - X_j$. When $(X_i, X_j) \in \boldsymbol{E}$ and $(X_j, X_i) \notin \boldsymbol{E}$ then X_i and X_j are connected by a *directed edge* (also referred to as an *arc*) and we write $X_i \to X_j$. When $(X_i, X_j) \notin \boldsymbol{E}$ and $(X_j, X_i) \notin \boldsymbol{E}$ we write $X_i \not- X_j$. When two vertices are joined by a directed or an undirected edge, the two vertices are *adjacent*.

When $X_i \to X_j$ then X_i is a *parent* of X_j, and X_j is a *child* of X_i. All parents of X_i belong to the parent set $\boldsymbol{X}_{pa(i)}$, where the indices of the parent vertices are returned by the function $pa(i)$. Similarly all children of X_i belong to the child set $\boldsymbol{X}_{ch(i)}$, where the indices are returned by $ch(i)$. A graph is *empty* if the set $\boldsymbol{E}$ is empty, and it is *complete* if any pairs of vertices are adjacent.

A graph where all edges are directed is called a *directed graph*, and a graph without directed edges is called an *undirected graph*. A *path* of length n between X_i and X_j is a sequence of distinct vertices $X_i = X^{(1)}, X^{(2)}, \ldots, X^{(n)} = X_j$, where $X^{(k-1)}$ and $X^{(k)}$ are adjacent, $k = 2, \ldots, n$. For a *directed path* it must hold that $X^{(k-1)} \to X^{(k)}$, i.e., every edge on the path in the directed graph points in the same forward direction.

In an undirected graph with $\boldsymbol{U}, \boldsymbol{S}, \boldsymbol{Y} \subset \boldsymbol{X}$, the set $\boldsymbol{S}$ *separates* $\boldsymbol{U}$ *from* $\boldsymbol{Y}$ if every path between vertices from $\boldsymbol{U}$ and $\boldsymbol{Y}$ intersects at least one vertex in $\boldsymbol{S}$.

A *directed cycle* is a directed path where begin and end vertices are the same, $X^{(1)} = X^{(n)}$. A *directed acyclic graph* (DAG) is a directed graph with no directed cycles. The *ancestors*, $\boldsymbol{X}_{an(i)}$, of X_i are all vertices that precede X_i on directed paths passing/to X_i. The *ancestral set* of a set $\boldsymbol{S}$ is $\bigcup_{X_i \in \boldsymbol{S}} \boldsymbol{X}_{an(i)}$. The *descendants*, $\boldsymbol{X}_{de(i)}$, of X_i are all vertices that succeed X_i on directed paths starting from X_i. The *non-descendants* is $\boldsymbol{X}_{nd(i)} = \boldsymbol{X} \setminus (\boldsymbol{X}_{de(i)} \cup \{X_i\})$.

A *serial connection* is $X_i \to X_k \to X_j$, and a *diverging connection* is $X_i \leftarrow X_k \to X_j$. A *converging connection* is $X_i \to X_k \leftarrow X_j$ where X_k is called a *collider*, and if additionally $X_i \neq X_j$ the connection is called a *v-connection*. A DAG without v-connections is called *moral* (the parents of X_k are "married" via an arc). A DAG can *be moralised* by marrying the offending parents of common children via an undirected edge, followed by dropping directions of all arcs. The *moralised* graph is thus an undirected graph.

3. Markov properties

We now associate random variables $\boldsymbol{X} = \{X_1, \ldots, X_p\}$ with the vertices of a graph, and use the graph to encode conditional independence restrictions for a joint probability distribution $\Pr(\boldsymbol{X})$. The so-called *Markov properties* specify how these independence statements relate to the graph: how to (en)code them and how to extract them.

A number of Markov properties can be ascribed to graphs. There are properties for DAGs and for undirected graphs. We will concentrate on

the three Markov properties for DAGs: The *directed pairwise* (DP), *directed local* (DL) and *the directed global* (DG) Markov properties. They are defined as follows:

DP For any non-adjacent vertices X_i and X_j with $X_j \in \boldsymbol{X}_{nd(i)}$, it holds that:

$$X_i \perp\!\!\!\perp X_j | \boldsymbol{X}_{nd(i)} \setminus \{X_j\}$$

Hence, X_i is independent of a non-descendant X_j given the remaining non-descendants, unless X_i and X_j are directly joined.

DL For any X_i it holds that:

$$X_i \perp\!\!\!\perp \boldsymbol{X}_{nd(i)} \setminus \boldsymbol{X}_{pa(i)} | \boldsymbol{X}_{pa(i)}$$

Hence, X_i is independent of its non-descendants given its parents.

DG For any disjoint sets $\boldsymbol{U}, \boldsymbol{S}, \boldsymbol{Y} \subseteq \boldsymbol{X}$ where $\boldsymbol{S}$ separates $\boldsymbol{U}$ from $\boldsymbol{Y}$ in the moralised graph of the ancestral set containing $\boldsymbol{U} \cup \boldsymbol{S} \cup \boldsymbol{Y}$, it holds that:

$$\boldsymbol{U} \perp\!\!\!\perp \boldsymbol{Y} | \boldsymbol{S}$$

As an example consider the DAG in figure 2.1, top. If we want to investigate whether $X_1 \perp\!\!\!\perp X_6 | \{X_3, X_4\}$ the graph containing the ancestral set of X_1, X_6, X_3, X_4 is formed, and the graph is moralised. The undirected graph in figure 2.1, bottom, left, is the result. From this we see that any path between X_1 and X_6 passes X_3, and the independence statement is correct.

To check if $X_1 \perp\!\!\!\perp X_4 | X_5$, the graph containing the ancestral set of X_1, X_4, X_5 is formed and moralised resulting in the graph in figure 2.1, bottom, right. From this we see that the independence statement does not hold: there is a path from X_1 to X_4 via X_3 that does not pass X_5.

The *d-separation* (DS) criterion by Pearl and Verma, 1987 provides and alternative to the directed global Markov property. Given $\boldsymbol{S} \subseteq \boldsymbol{X}$ and any pair of vertices $X_i, X_j \notin \boldsymbol{S}$ a path between X_i and X_j is *active* if:

1 Every non-collider in the path is not in $\boldsymbol{S}$.

2 Every collider in the path is in $\boldsymbol{S}$ or has descendants in $\boldsymbol{S}$.

If $\boldsymbol{S}$ creates an active path between X_i and X_j then they cannot be conditionally independent given $\boldsymbol{S}$. When the path is *not* active, the path is *blocked* by $\boldsymbol{S}$.

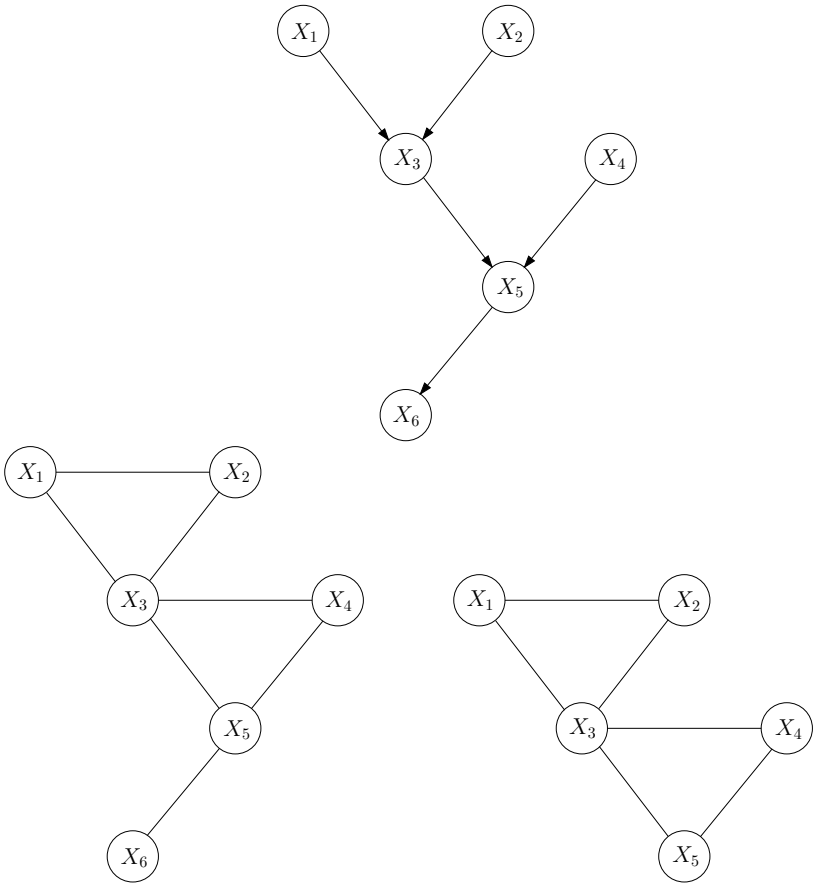

Figure 2.1. The DG Markov property. *Top:* a DAG model. *Bottom:* two moralised graphs.

DS For disjoint sets $\boldsymbol{U}, \boldsymbol{S}, \boldsymbol{Y} \subseteq \boldsymbol{X}$, $\boldsymbol{U}$ is *d-separated* from $\boldsymbol{Y}$ by $\boldsymbol{S}$ if every path between vertices in $\boldsymbol{U}$ and $\boldsymbol{Y}$ is blocked by $\boldsymbol{S}$.

This reading captures a notion of *flow* between variables when the paths between them aren't blocked.

Lauritzen et al., 1990 show that DS is equivalent to DG. Additionally it turns out that DG is equivalent to DL which in turn imply DP:

$$\text{DS} \Leftrightarrow \text{DG} \Leftrightarrow \text{DL} \Rightarrow \text{DP}$$

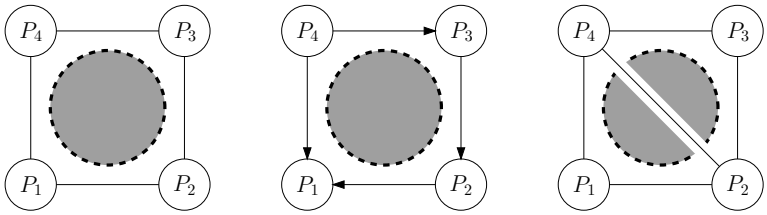

Figure 2.2. The "Yawning philosophers problem": The shaded circle in the middle represents the pillar, and $P_1, \ldots, P_4$ are the yawning philosophers standing around it. The arcs/edges are the lines of sight. In the picture to the right, the pillar necessarily has to be ignored (!) as a consequence of a DAG model representation.

A probability distribution $\Pr(\cdot)$ obeying the DG Markov property with respect to a DAG, factorises in the following way:

$$\Pr(\boldsymbol{X}) = \prod_{i=1}^{p} \Pr(X_i | \boldsymbol{X}_{pa(i)}) \tag{2.1}$$

It is not necessarily so that all conditional independences of a given probability distribution can be captured by a DAG model, or, equivalently, that the distribution can be factorised as in eq. 2.1. When the independences induced by the DAG model are a subset of the independences holding for a given joint distribution, we say that the DAG model is an *I-map* (independence map) of the probability distribution. There may be more independences holding in the probability distribution than what can be read off the DAG.

As an example consider the following "Yawning philosophers problem": Suppose that $P_i, i = 1, \ldots, 4$ denotes the binary random variable representing whether philosopher i is yawning or not. For mammals it is well-known that the probability of yawning is increased when one *sees* other mammals yawning. Position the philosophers around a Greek pillar starting with P_1 followed by P_2 etc., such that P_i cannot see the philosopher on the other side of the pillar but only the two philosophers on his left and right side; figure 2.2, left, depicts this. Hence for P_i it holds that he has direct visual contact with his two neighbours. The probability of P_i yawning depends on his neighbours, but is independent on the philosopher on the other side given the yawing state of his neighbours. Hence the following independence statements hold: $P_1 \perp\!\!\!\perp P_3 | \{P_2, P_4\}$, $P_2 \perp\!\!\!\perp P_4 | \{P_1, P_3\}$.

No DAG model can capture the two independence assumptions simultaneously. The associations between the four variables via four directed arcs will necessarily introduce a v-connection because no cycles are allowed; see figure 2.2, middle, for an example. The moralised version of this DAG will join either $P_4 - P_2$ or $P_1 - P_3$; see figure 2.2, right. Using the DG Markov property this in effect means that one of the conditional independences actually holding cannot be read off the DAG. In more general terms, when a v-connection is unavoidable, all independence statements that hold cannot be encoded simultaneously; e.g., for many grid like structures where each intersection represents a random variable, the "yawning philosophers problem" exists.

We say that a DAG model is a *D-map* (dependence map) of a given joint probability distribution, if all independences that hold for the joint distribution can be read off the DAG model. A DAG model is a *P-map* (perfect map) of a probability distribution if it is both an I-map and a D-map of the joint distribution; equivalently we say that the probability distribution is *faithful* with respect to the DAG model. In that case not only does the *absence* of edges encode *independences*, but the *presence* of edges also encode *dependences*.

It remains unclear whether most "real-life" distributions or processes are faithful with respect to DAG models. Strictly speaking this means that when we learn DAG models from empirical observations, as we will in Chapter 3, we may not interpret the presence of arcs in a DAG model as dependences. However, for large DAG models some parts of the DAG model may be a perfect map and other parts may not. Admittedly, although this remains rather vague, it is our hope that a substantial part of the DAG model indeed does capture the dependences and only a few dependences we read off are wrong. We would like to stress that the underlying assumption here is that the objective is to interpret the DAG model in terms of the (in)dependences it encodes. From the perspective of prediction one may not be interested in interpreting the DAG model at all, but merely use it as a concise way of representing a probability distribution in terms of eq. 2.1.

3.1 The Markov blanket

An important notion is that of the *Markov blanket* of a random variable X_i. This term is used throughout this thesis, as it plays an important role in many learning contexts. The interesting thing about this set

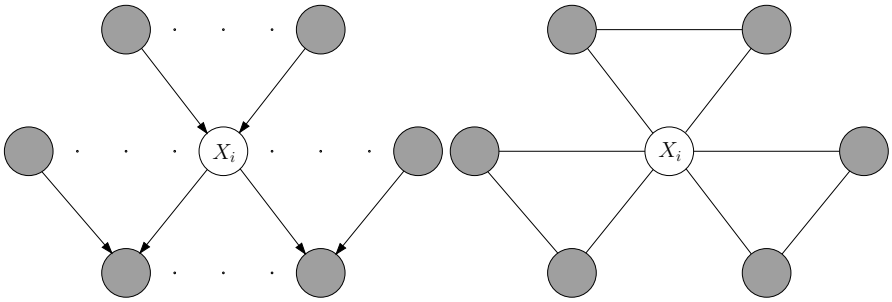

Figure 2.3. The Markov blanket of X_i. *Left:* the DAG model. *Right:* the moralised graph.

is that X_i is independent of all other variables given its Markov blanket; the blanket so to speak shields off influence from outside.

The Markov blanket of X_i is:

$$\boldsymbol{X}_{mb(i)} = \boldsymbol{X}_{pa(i)} \cup \boldsymbol{X}_{ch(i)} \bigcup_{j \in ch(i)} \boldsymbol{X}_{pa(j)} \setminus \{X_i\}$$

i.e., the parents, the children, and the parents of the children. The graph in figure 2.3, left, illustrates the Markov blanket of X_i; the shaded vertices belong to the Markov blanket of X_i. The graph in figure 2.3, right, is a part of the moralised graph of the DAG. From this graph it is easy to see that all paths to X_i necessarily have to pass a shaded vertex. Using the DG Markov property we then have that $X_i \perp\!\!\!\perp \boldsymbol{X} \setminus (\boldsymbol{X}_{mb(i)} \cup \{X_i\}) | \boldsymbol{X}_{mb(i)}$. It follows that the Markov blanket consists of those variables that are immediately relevant for prediction of X_i.

3.2 Equivalence of DAGs

Different DAGs can represent the same set of conditional independence restrictions; in that case we say that DAGs are *equivalent*. The DAG models are partitioned into equivalence classes. Two DAGs are equivalent if the following two conditions both hold (Verma and Pearl, 1990):

1 By dropping directions, the resulting undirected graphs are the same.

2 The two DAGs have the same v-connections.

Edges of DAGs that have the same direction in all DAGs in a particular equivalence class are *compelled*; the other edges are *reversible*. For

instance from the DAG $X_1 \rightarrow X_2 \rightarrow X_3$ we can read off $X_1 \perp\!\!\!\perp X_3 | X_2$, but the two alternatives $X_1 \leftarrow X_2 \rightarrow X_3$ and $X_1 \leftarrow X_2 \leftarrow X_3$ also encode $X_1 \perp\!\!\!\perp X_3 | X_2$; all these DAGs are equivalent because by dropping directions they are the same undirected graph, and they all have the same v-connections (none). In this example both arcs are reversible.

A *canonical* representation of an equivalence class is obtained via a partially directed-undirected graph. For all reversible edges the direction is dropped and for compelled edges the direction is kept. The resulting graph is referred to as an *essential graph* (Andersson et al., 1997). For $X_1 \rightarrow X_2 \rightarrow X_3$, the essential graph is $X_1 - X_2 - X_3$. Given a DAG, the essential graph of its equivalence class can be obtained in polynomial time via an algorithm given in Chickering, 1995.

Given two different DAGs, m_i and m_j, in the same equivalence class, it is possible to transform m_i into m_j, through a series of elementary graph operations, without leaving the equivalence class. The notion of *covered arc reversal* is required for this purpose (Chickering, 1995). An arc $X_i \rightarrow X_j$ is *covered* if the vertices pertaining to the arc have the same parent set, i.e., $\boldsymbol{X}_{pa(i)} = \boldsymbol{X}_{pa(j)} \backslash \{X_i\}$. Reversing a covered arc will not alter the v-connections, and by dropping directions, the undirected graph will remain unchanged. This means that the DAG obtained by a single covered arc reversal is equivalent to the initial DAG. Furthermore, there exists a sequence of DAGs, $m_i = m^{(1)}, \ldots, m^{(n)} = m_j$, where $m^{(k+1)}$ is obtained from $m^{(k)}, k = 1, \ldots, n-1$ by reversing a covered arc in $m^{(k)}$. Consequently, all DAGs of an equivalence class can be reached by repeated covered arc reversals. Note that a compelled arc can never be covered, but also that a reversible arc is not necessarily covered for a given DAG in the sequence; reversible arcs are only covered in a given graph in the equivalence class. In the example $X_1 \rightarrow X_2 \rightarrow X_3$ arc $X_1 \rightarrow X_2$ is covered, but $X_2 \rightarrow X_3$ is not even though $X_2 \rightarrow X_3$ is reversible. Reversing the covered arc $X_1 \rightarrow X_2$ we get the equivalent DAG, $X_1 \leftarrow X_2 \rightarrow X_3$. Now both arcs are covered, and by reversing $X_2 \rightarrow X_3$ we obtain another equivalent DAG, $X_1 \leftarrow X_2 \leftarrow X_3$.

The fact that many arcs are reversible suggests that arcs should not be interpreted as causal influences. From a statistical point of view equivalent DAGs are *statistically indistinguishable*, and in order to assign any causal semantics to arcs, certain assumptions definitely have to be made. We refer to Pearl, 2000; Spirtes et al., 2000 for treatment of these matters. In this thesis we assign no particular semantics to edges except for the conditional independence restrictions that they encode via the Markov properties.

4. Bayesian networks

Up until now, the focus has been on the conditional independences and the factorisation of a joint probability distribution according to some DAG model. In the remainder we distinguish the qualitative constraints dictated by the DAG model from the quantitative part consisting the local conditional probabilities defined "along the arcs" of the DAG model. We use the term *Bayesian Network* (BN) to denote the pair $(m, \boldsymbol{\theta})$ (Pearl, 1988) consisting of:

1 The DAG model $m = (\boldsymbol{X}, \boldsymbol{E})$ encoding conditional independence statements. X_i is *discrete* random variable associated with a vertex of m.

2 The *parameter* of m denoted by $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_p)$, where $\boldsymbol{\theta}_i$ consists of the local probabilities $\theta_{X_i|\boldsymbol{X}_{pa(i)}} = \Pr(X_i|\boldsymbol{X}_{pa(i)})$ in eq. 2.1.

Moreover, we make m and $\boldsymbol{\theta}$ explicit, and write eq. 2.1 as:

$$\Pr(\boldsymbol{X}|m, \boldsymbol{\theta}) = \prod_{i=1}^{p} \Pr(X_i|m, \boldsymbol{X}_{pa(i)}, \boldsymbol{\theta}_i) = \prod_{i=1}^{p} \theta_{X_i|\boldsymbol{X}_{pa(i)}} \tag{2.2}$$

Via the usual algebraic laws of probability theory, any marginal or conditional probability (distribution) can be computed. However, by exploiting the independences induced by the DAG, the efficiency of these computations can be improved. Several exact inference algorithms have been developed that use the DAG directly as computational backbone for these calculations, such as the method described in Pearl, 1986. Other inference algorithms use the moralised graph as a departure point plus some additional transformations to perform these computations, such as the algorithm by Lauritzen and Spiegelhalter, 1988 and Jensen et al., 1990. In general however, inference in a BN is NP-hard (Cooper, 1990), and, even for feasible problems, inference is considered a computationally demanding operation.

5. Bayesian network specification

We can either specify a BN by hand, learn it from data, or, combine both approaches, i.e., use prior knowledge and complement this knowledge with what can be learned from data. The Bayesian statistical paradigm offers a principled approach for dealing with knowledge updating; it describes how to combine and update expert knowledge with observational data. It provides a sound framework for representing

and reasoning with uncertainty and imprecision. Within the scope of *learning* these concepts are of great importance; the finite sample size and missing data are sources that introduce uncertainty or imprecision with respect to what is subject to learning.

Another advantage of the Bayesian approach is that it is typically very clear about the different assumptions made by explicitly conditioning on these assumptions; in order to get rid of them, they have to be marginalised out again.

A common misconception is that the Bayesian statistical approach is inherent to the Bayesian network formalism; this is not the case. In fact, when specifying a BN, by hand or otherwise, the classical (frequentist) approach is usually taken. We refer to Gelman et al., 2004 for more on the Bayesian approach to data analysis in general.

5.1 Bayesian parameter specification

This section serves as an illustration of how "to be a Bayesian". In particular we show how the parameter of a BN is treated from a Bayesian point of view. Additionally this section is a departure point for the next chapter about learning BNs.

We assume that m is fixed (given), but that the parameter is subject to uncertainty and is considered as a random variable $\boldsymbol{\Theta}$. The uncertainty of $\boldsymbol{\Theta}$ is captured by a distribution $\Pr(\boldsymbol{\Theta}|m)$. This distribution conveys all information we have about $\boldsymbol{\Theta}$.

Since the parameter is not *assumed given* anymore as in eq. 2.2 but is subject to learning, it is integrated out with respect to $\Pr(\boldsymbol{\Theta}|m)$:

$$\Pr(\boldsymbol{X}|m) = \int \Pr(\boldsymbol{X}, \boldsymbol{\theta}|m) d\boldsymbol{\theta} = \int \Pr(\boldsymbol{X}|m, \boldsymbol{\theta}) \cdot \Pr(\boldsymbol{\theta}|m) d\boldsymbol{\theta} \qquad (2.3)$$

The equation is a kind of "averaging" over possible values that $\boldsymbol{\Theta}$ can take on, depending on the uncertainty of the possible parameter values captured by $\Pr(\boldsymbol{\Theta}|m)$. At this point we see that $\Pr(\boldsymbol{\Theta}|m)$ is conditional on the fixed model m only; it is the *prior* distribution before having been presented with any information that influences the "knowledge" about the parameter.

If we have that:

$$\Pr(\boldsymbol{\Theta}|m) = \prod_{i=1}^{p} \Pr(\boldsymbol{\Theta}_i|m) \qquad (2.4)$$

i.e., that parameters $\boldsymbol{\Theta}_i$ are *globally independent* (Spiegelhalter and Lauritzen, 1990) we get from eq. 2.3 and the BN factorisation in eq. 2.2:

$$\begin{aligned}\Pr(\boldsymbol{X}|m) &= \int \prod_{i=1}^{p} \Pr(X_i|\boldsymbol{X}_{pa(i)}, m, \boldsymbol{\theta}_i) \cdot \Pr(\boldsymbol{\theta}_i|m) d\boldsymbol{\theta} \\ &= \prod_{i=1}^{p} \mathrm{E}[\boldsymbol{\Theta}_i]_{X_i|\boldsymbol{X}_{pa(i)}} \\ &= \prod_{i=1}^{p} \Pr(X_i|\boldsymbol{X}_i, \boldsymbol{\theta}_i = \mathrm{E}[\boldsymbol{\Theta}_i], m) \qquad (2.5)\end{aligned}$$

where $\mathrm{E}[\boldsymbol{\Theta}_i] = \int \boldsymbol{\theta}_i \cdot \Pr(\boldsymbol{\theta}_i|m) d\boldsymbol{\theta}_i$. A consequence of this assumption is that in order to *exploit* what is (currently) known about the parameter and the associated uncertainty for computing any conditional or marginal probability of $\boldsymbol{X}$, we should use the expected parameter values per vertex with respect to the parameter distributions that capture the parameter uncertainty.

Furthermore, from the integrand in eq. 2.3 we see that:

$$\Pr(\boldsymbol{X}, \boldsymbol{\Theta}|m) = \prod_{i=1}^{p} \Pr(X_i|m, \boldsymbol{X}_{pa(i)}, \boldsymbol{\Theta}_i) \cdot \Pr(\boldsymbol{\Theta}_i|m)$$

i.e., $\boldsymbol{\Theta}_i$ can be considered an extra parent of X_i in m; the result is an augmented BN. Using the DG Markov property, it is then possible to check which variables influence the parameter $\boldsymbol{\Theta}_i$. Hence, in order to update the parameters once additional observations are presented, the influence restrictions determine how this evidence is distributed towards these parameters, i.e., what impact new information has on the distribution on the parameter space. For instance, given some evidence $\boldsymbol{O} \subset \boldsymbol{X}$ in the form of a new incoming "fact", we are interested in the joint distribution on $\boldsymbol{X}$ *given* this new fact, $\Pr(\boldsymbol{X}|m, \boldsymbol{O}) = \prod_{i=1}^{p} \mathrm{E}[\boldsymbol{\Theta}_i|\boldsymbol{O}]$. Here the expectation is taken with respect to the *posterior* parameter distribution $\Pr(\boldsymbol{\Theta}|m, \boldsymbol{O})$, i.e., the parameter distribution conditional on $\boldsymbol{O}$. The question that can be answered via the DG Markov property is then, how the posterior distribution of $\boldsymbol{\Theta}$ depends on $\boldsymbol{O}$.

In particular observe that $\boldsymbol{\Theta}_i$ is independent of all other vertices given X_i and $\boldsymbol{X}_{pa(i)}$, because this conditioning set is exactly the Markov blanket of $\boldsymbol{\Theta}_i$ (the vertex has no parents). Hence, given that X_i and $\boldsymbol{X}_{pa(i)}$ are in $\boldsymbol{O}$, we have that $\Pr(\boldsymbol{\Theta}_i|m, \boldsymbol{O}) = \Pr(\boldsymbol{\Theta}_i|m, X_i, \boldsymbol{X}_{pa(i)})$.

In figure 2.4, left, an example of an augmented BN is shown. Suppose that we observe $\boldsymbol{O} = (X_4, X_5)$, and that we wonder if this has an impact

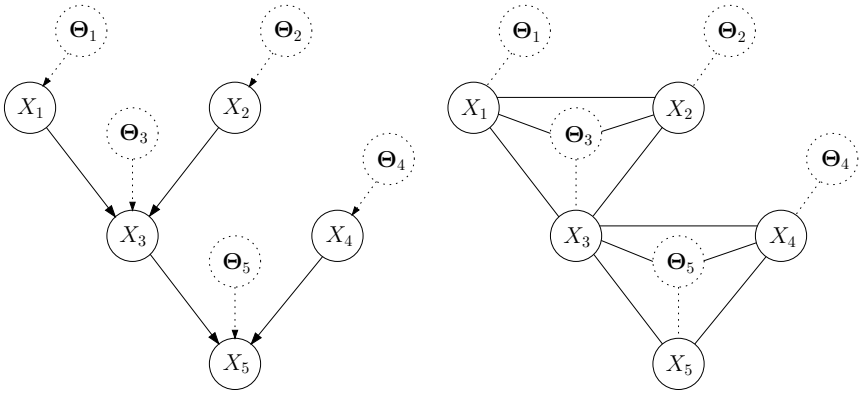

Figure 2.4. The Bayesian approach to parameter learning. *Left:* an augmented BN. *Right:* the moralised graph.

on the posterior distribution for $\boldsymbol{\Theta}_3$. We create the moralised graph in figure 2.4, right. From this graph we can read off that $\boldsymbol{\Theta}_3$ indirectly depends on X_5. This is also the case for X_4 even though it is far from being "local" to $\boldsymbol{\Theta}_3$. Had X_3 been part of our observation, then influence from X_4 and X_5 would have been blocked. The fact that this is not the case means that the posterior parameter distribution for $\boldsymbol{\Theta}_3$ is updated via a BN inference method, that is, $\boldsymbol{O} = (X_4, X_5)$ gives rise to a certain amount of information that is added to our knowledge. This example illustrates that $\boldsymbol{X}$ and $\boldsymbol{\Theta}$ are essentially treated in the same way.

In the ideal situation, all assumptions should be regarded as random variables and belong on the conditional side of a distribution. Once we decide to marginalise them out we are forced to quantify their uncertainty in terms of distributions as in eq. 2.3. When learning BNs from data, usually only the quantities that depend on the data sample are treated as random variables. In this section only the parameter of a BN was treated in a Bayesian fashion; in Chapter 4, Section 3 we go one step further, and treat the edges of the BN DAG model as random variables.

In this regard it is interesting to observed that "learning" is intrinsic to Bayesian statistics. When confronted with new observations, the Bayesian applies an updating rule based on Bayes' law which revises existing distributions thereby capturing a notion of "learning from experience"; our belief is dynamically changed as more information is absorbed, and at any given time, we are able to exploit what has been gathered.

Chapter 3

LEARNING BAYESIAN NETWORKS FROM DATA

In this chapter we discuss how to learn Bayesian network models and the parameters from data. In Chapter 2, Section 5.1 the Bayesian approach to "learning from experience" was introduced. In this chapter we continue along these lines, and we apply the Bayesian updating approach given a whole set of observations. There are however alternatives to the Bayesian approach; some of these alternatives are treated as well, though in less detail. For model learning we do not discuss the so-called constraint-based approaches based on independence tests that are conducted on the data, from which the BN models are derived (Spirtes et al., 2000; Bromberg et al., 2006).

1. Learning from data

The goal is to learn from an independent and identically distributed (iid) data sample $\boldsymbol{D} = (\boldsymbol{D}^1, \ldots, \boldsymbol{D}^c)$, consisting of c cases (or records). Each record $\boldsymbol{d}^j = (x_1^j, \ldots, x_p^j)$, contains instantiations x_i^j of variable X_i for $i = 1, \ldots, p$ for the jth case.

We reduce the data sample to contingency tables representing vectors with observational counts distributed according to a *product multinomial* distribution parameterised by $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_p)$ where $\boldsymbol{\theta}_i$ consists of the local parameters $\theta_{X_i|\boldsymbol{X}_{pa(i)}}$ that determine the probability of a case falling into a particular cell associated with vertex X_i. The cell counts are extracted from the contingency tables using the function $n(\cdot)$, e.g., $n(x_i, \boldsymbol{x}_{pa(i)})$ returns the number of occurrences in the data sample of $X_i = x_i$ *and* $\boldsymbol{X}_{pa(i)} = \boldsymbol{x}_{pa(i)}$. Note that by maintaining the counts only,

we seem to lose information about the ordering of the data records. However, there is no information loss, the iid assumption implies that the ordering is completely random.

The *likelihood function* of the product multinomial distribution is:

$$\mathcal{L}(m, \boldsymbol{\theta}|\boldsymbol{d}) \propto \Pr(\boldsymbol{d}|m, \boldsymbol{\theta}) = \prod_{i=1}^{p} \prod_{\boldsymbol{x}_{pa(i)}} \left(\prod_{x_i} \theta_{x_i|\boldsymbol{x}_{pa(i)}}^{n(x_i, \boldsymbol{x}_{pa(i)})} \right) \quad (3.1)$$

$$= \prod_{j=1}^{c} \prod_{i=1}^{p} \Pr(x_i^j|m, \boldsymbol{x}_{pa(i)}^j, \boldsymbol{\theta}_i) \quad (3.2)$$

Henceforth, we ignore the proportionality factor, and refer to $\Pr(\boldsymbol{d}|m, \boldsymbol{\theta})$ instead of $\mathcal{L}(m, \boldsymbol{\theta}|\boldsymbol{d})$ as the likelihood function. The expression in eq. 3.2 explicitly shows that each record j is an independent realisation from the BN $(\boldsymbol{\theta}, m)$.

The likelihood function plays a crucial role in learning BNs. It describes the process giving rise to the data in terms of the model and parameter.

2. Learning parameters

Learning the parameter of a BNs is computationally less demanding than learning the DAG model. For complete data, iteration or approximation is not required. Closed version solutions are available as functions of the counts, $n(\cdot)$.

2.1 The Maximum-Likelihood approach

The likelihood function $\Pr(\boldsymbol{d}|m, \boldsymbol{\theta})$ should be read as "the likelihood of $(m, \boldsymbol{\theta})$ given $\boldsymbol{d}$". Following this reading, for a fixed model, one approach to learning the parameter, is to maximise the likelihood with respect to $\boldsymbol{\theta}$. The intuition behind this, is that the parameter hypothesis that has the maximum support from data, is the best hypothesis possible:

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \Pr(\boldsymbol{d}|m, \boldsymbol{\theta})$$

For the multinomial distribution, it is easy to determine this estimate. The first moment of the multinomial distribution is:

$$\mathrm{E}[n(x_i, \boldsymbol{x}_{pa(i)})|\boldsymbol{\theta}_i, m] = \theta_{x_i|\boldsymbol{x}_{pa(i)}} \cdot \sum_{x_i} n(x_i, \boldsymbol{x}_{pa(i)}) \quad (3.3)$$

The multinomial distribution belongs to the *regular exponential family*, i.e., it may be written in a particular functional form. The exponential

family is a function of the so-called *sufficient statistics* that are functions of the data, $\boldsymbol{T}(\boldsymbol{d})$; given those statistics, $\boldsymbol{d}$ brings no additional information about $\boldsymbol{\theta}$ (Casella and Berger, 2002). For the multinomial distribution $\boldsymbol{T}(\boldsymbol{d})$ coincides with the counts, $n(x_i, \boldsymbol{x}_{pa(i)})$. For the exponential family, the Maximum-Likelihood (ML) estimator is found by solving $\mathrm{E}[\boldsymbol{T}(\boldsymbol{d})|\boldsymbol{\theta}, m]$ with respect to $\boldsymbol{\theta}$ (Cox and Hinkley, 1974; Lehmann and Casella, 2001). Thus for the multinomial, equating the cell counts to their expectations using eq. 3.3, it follows that the ML estimators are:

$$\hat{\theta}_{x_i|\boldsymbol{x}_{pa(i)}} = \frac{n(x_i, \boldsymbol{x}_{pa(i)})}{\sum_{x_i} n(x_i, \boldsymbol{x}_{pa(i)})} \tag{3.4}$$

2.2 The Bayesian approach

The Bayesian does not admit to the reading "best support given data", because it implies that our knowledge about $\boldsymbol{\theta}$ is fully determined by $\boldsymbol{d}$, irrespectively of what is known beyond $\boldsymbol{d}$. Any common, accumulated or *prior* knowledge is essentially joined together with $\boldsymbol{d}$ forming a batch using ad-hoc principles or methods that are less "elegant" and principled than one could hope for.

As discussed in Chapter 2, Section 5.1, the Bayesian treats the parameter as a random variable, and defines a distribution on the parameter space. This prior distribution captures the uncertainty with respect to the parameter. As we saw in Chapter 2, Section 5.1, the expectation of the posterior distribution is the logical choice when calculating any marginal or conditional distribution for $\boldsymbol{X}$. In general however, the Bayesian is interested in much more than merely the expectation of the posterior distribution; the *entire* distribution is interesting. Specifically, ultimately we are interested in the posterior distribution $\Pr(\boldsymbol{\Theta}|m, \boldsymbol{d})$, since this contains everything we know about $\boldsymbol{\Theta}$.

The actual updating of prior to posterior distribution goes via Bayes' law. Bayes' law combines the information contained in the likelihood function with the prior parameter distribution, and transforms it into a posterior parameter distribution:

$$\Pr(\boldsymbol{\Theta}|m, \boldsymbol{d}) = \frac{\Pr(\boldsymbol{d}|m, \boldsymbol{\Theta}) \cdot \Pr(\boldsymbol{\Theta}|m)}{\int \Pr(\boldsymbol{d}|m, \boldsymbol{\theta}) \cdot \Pr(\boldsymbol{\theta}|m) d\boldsymbol{\theta}} \tag{3.5}$$

$$\propto \prod_{i=1}^{p} \prod_{\boldsymbol{x}_{pa(i)}} \left(\prod_{x_i} \theta_{x_i|\boldsymbol{x}_{pa(i)}}^{n(x_i, \boldsymbol{x}_{pa(i)})} \right) \cdot \Pr(\boldsymbol{\Theta}|m) \tag{3.6}$$

The denominator in eq. 3.5 is the *normalising term*. Alternatively, instead of considering $\boldsymbol{d}$ as a single batch, we may update the posterior on a per case basis; first we are presented with $\boldsymbol{d}^1$ resulting in the posterior $\Pr(\boldsymbol{\Theta}|m, \boldsymbol{d}^1)$. This posterior acts as the prior for the next update when $\boldsymbol{d}^2$ is presented, resulting in the new posterior $\Pr(\boldsymbol{\Theta}|m, \boldsymbol{d}^1, \boldsymbol{d}^2)$, and so on. In the end, this will give us the same posterior as $\Pr(\boldsymbol{\Theta}|m, \boldsymbol{d})$; facts have accumulated over time. For complete data the ordering of the records is unimportant.

Although not necessarily a requirement, for computational convenience it is advantageous if the prior and posterior distribution have the same functional form with a simple update rule. Such a distribution is called a *conjugate prior*. From a theoretical point of view conjugacy may seem as a severe constraint, but unfortunately it is a necessary one from a practical point of view.

Previously in Chapter 2, Section 5.1 the notion of *global parameter independence* was introduced as the decomposition given in eq. 2.4, that is, the assumption that the conditional probabilities for each child variable X_i can be specified or learned separately from each other. If in addition to global independence we also have *local parameter independence* (Spiegelhalter and Lauritzen, 1990), that is:

$$\Pr(\boldsymbol{\Theta}_i|m) = \prod_{\boldsymbol{x}_{pa(i)}} \Pr(\boldsymbol{\Theta}_{X_i|\boldsymbol{x}_{pa(i)}}|m)$$

when combined with global independence, the overall parameter distribution becomes:

$$\Pr(\boldsymbol{\Theta}|m) = \prod_{i=1}^{p} \Pr(\boldsymbol{\Theta}_i|m) = \prod_{i=1}^{p} \prod_{\boldsymbol{x}_{pa(i)}} \Pr(\boldsymbol{\Theta}_{X_i|\boldsymbol{x}_{pa(i)}}|m)$$

which states that each parameter distribution conditional on a parent configuration, can be updated independently of each other.

Similar to the requirement of conjugacy, assuming parameter independence is from a computational point of view the only feasible thing to do. However, it is not an entirely unrealistic assumption. From a domain perspective it makes sense that parameter specification can be constrained to particular "regions" of a BN. Perhaps the most unrealistic assumption is therefore local parameter independence. After all these parameters pertain to the same child/parent vertices, and therefore one may argue that it is unreasonable that they can be specified independently of each other.

Since we are dealing with a product multinomial, we see from eq. 3.1 that the likelihood of the product multinomial is simply a product of likelihoods for each multinomial distribution as well. In fact the decomposition per vertex and parent set configuration coincides with the decomposition of the overall parameter distribution of $\boldsymbol{\Theta}$ under global and local parameter independence. The conjugate prior for the multinomial is the *Dirichlet* distribution. Hence we define a Dirichlet distribution for each multinomial:

$$\Pr(\boldsymbol{\Theta}_{X_i|\boldsymbol{x}_{pa(i)}}|m) = \mathrm{Dir}(\boldsymbol{\Theta}_{X_i|\boldsymbol{x}_{pa(i)}}|\boldsymbol{\alpha}) = \prod_{x_i} \Theta_{x_i|\boldsymbol{x}_{pa(i)}}^{\alpha(x_i,\boldsymbol{x}_{pa(i)})-1} \cdot Z(\boldsymbol{x}_{pa(i)}, \boldsymbol{\alpha})$$

where $\boldsymbol{\alpha} > \mathbf{0}$ is the vector of prior hyper parameters of the Dirichlet.

The *normalising factor* is:

$$Z(\boldsymbol{x}_{pa(i)}, \boldsymbol{\alpha}) = \frac{1}{\int \prod_{x_i} \theta_{x_i|\boldsymbol{x}_{pa(i)}}^{\alpha(x_i,\boldsymbol{x}_{pa(i)})-1} d\boldsymbol{\theta}_{X_i|\boldsymbol{x}_{pa(i)}}} = \frac{\Gamma(\sum_{x_i} \alpha(x_i, \boldsymbol{x}_{pa(i)}))}{\prod_{x_i} \Gamma(\alpha(x_i, \boldsymbol{x}_{pa(i)}))} \tag{3.7}$$

Here $\Gamma(\cdot)$ is the gamma function. Note that the normalising factor is a function of the parent set of X_i and the hyper parameter $\boldsymbol{\alpha}$.

Applying Bayes' rule from eq. 3.6, it is easy to see that the posterior is a product Dirichlet:

$$\begin{aligned} \Pr(\boldsymbol{\Theta}|m, \boldsymbol{d}) &\propto \prod_{i=1}^{p} \prod_{\boldsymbol{x}_{pa(i)}} \prod_{x_i} \Theta_{x_i|\boldsymbol{x}_{pa(i)}}^{n(x_i,\boldsymbol{x}_{pa(i)})} \cdot \prod_{i=1}^{p} \prod_{\boldsymbol{x}_{pa(i)}} \prod_{x_i} \Theta_{x_i|\boldsymbol{x}_{pa(i)}}^{\alpha(x_i,\boldsymbol{x}_{pa(i)})-1} \\ &= \prod_{i=1}^{p} \prod_{\boldsymbol{x}_{pa(i)}} \prod_{x_i} \Theta_{x_i|\boldsymbol{x}_{pa(i)}}^{\alpha(x_i,\boldsymbol{x}_{pa(i)})+n(x_i,\boldsymbol{x}_{pa(i)})-1} \end{aligned} \tag{3.8}$$

where the normalising term for the posterior is updated with the sufficient statistics $\boldsymbol{n}$ from $\boldsymbol{d}$:

$$\prod_{i=1}^{p} \prod_{\boldsymbol{x}_{pa(i)}} Z(\boldsymbol{x}_{pa(i)}, \boldsymbol{\alpha} + \boldsymbol{n}) = \prod_{i=1}^{p} \prod_{\boldsymbol{x}_{pa(i)}} \frac{\Gamma(\sum_{x_i} \alpha(x_i, \boldsymbol{x}_{pa(i)}) + n(x_i, \boldsymbol{x}_{pa(i)}))}{\prod_{x_i} \Gamma(\alpha(x_i, \boldsymbol{x}_{pa(i)}) + n(x_i, \boldsymbol{x}_{pa(i)}))} \tag{3.9}$$

For sequential updating the same holds; depending on the configuration $\boldsymbol{d}^j$ only those statistics are increased that correspond to the observation in record j.

The prior hyper parameters, $\boldsymbol{\alpha}$, are often considered *imaginary counts* corresponding to cell counts prior to observing data. However, interpreting $\alpha(x_i, \boldsymbol{x}_{pa(i)}) = 0$ as non-informative in the sense that no prior

counts are available does not result in a posterior Dirichlet unless all $n(x_i, \boldsymbol{x}_{pa(i)}) > 0$. From an inference point of view assigning a small value > 0 to $\alpha(x_i, \boldsymbol{x}_{pa(i)})$ may be helpful in making summary statistics more stable. For instance for large models, relatively many cell counts are typically zero, in which case adding a small number of prior observations makes analysis easier.

It is worth noticing that for large samples, the prior counts will have a minimal impact on the posterior. For small samples, the prior counts do have a substantial impact, and should be chosen with some care.

To a Bayesian, the posterior distribution is the object of inference, and in some cases it may even be "the answer" itself. However, from a practical point of view, summary statistics of the distribution are often used to get at least a partial impression of the distribution. Denote $\alpha(x_i, \boldsymbol{x}_{pa(i)}) + n(x_i, \boldsymbol{x}_{pa(i)}) = s(x_i, \boldsymbol{x}_{pa(i)})$ and $\sum_{x_i} s(x_i, \boldsymbol{x}_{pa(i)}) = s(\boldsymbol{x}_{pa(i)})$. The expectation and the variance of the (posterior) Dirichlet are given by:

$$\mathrm{E}[\Theta_{x_i|\boldsymbol{x}_{pa(i)}}] = \frac{s(x_i, \boldsymbol{x}_{pa(i)})}{s(\boldsymbol{x}_{pa(i)})} \tag{3.10}$$

$$\mathrm{V}[\Theta_{x_i|\boldsymbol{x}_{pa(i)}}] = \frac{s(\boldsymbol{x}_{pa(i)}) \cdot s(x_i, \boldsymbol{x}_{pa(i)}) - s(x_i, \boldsymbol{x}_{pa(i)})^2}{s(\boldsymbol{x}_{pa(i)})^3 + s(\boldsymbol{x}_{pa(i)})^2} \tag{3.11}$$

Fixing the parameter expectations, we see that the variance decreases as $s(\boldsymbol{x}_{pa(i)})$ increases and the Dirichlet becomes more tight around the parameter expectations, indicating that the uncertainty decreases.

The multinomial likelihood and the Dirichlet are the same up to a constant when we subtract one from all the counts in the Dirichlet. The ML estimator for the multinomial is given in eq. 3.4 as a function of the counts $n(x_i, \boldsymbol{x}_{pa(i)})$. If we use the counts $s(x_i, \boldsymbol{x}_{pa(i)}) - 1$ instead, it follows that for $\alpha(x_i, \boldsymbol{x}_{pa(i)}) \geq 1$ that the mode of the Dirichlet is located at:

$$\frac{s(x_i, \boldsymbol{x}_{pa(i)}) - 1}{s(\boldsymbol{x}_{pa(i)}) - |\Omega_{X_i}|}$$

3. Learning models

The absence of arcs in the DAG m encodes assumptions about independences and therefore a complete DAG encodes no assumptions about independences at all—it is called the *saturated model*. A saturated model can capture any distribution for $\boldsymbol{X}$, because any minor interaction effects between variables can be modeled by joining variables. The more

dense the graph, the more detail can be captured. A model that is too dense will pick up spurious and noisy relationships in the data at hand. We seek the model that captures the interactions of *any* data set generated by the underlying process of $\boldsymbol{d}$, not a model that only captures all detail of $\boldsymbol{d}$ in particular. The term *overfitting* refers to the problem pertaining to learning models that fit the data in every detail versus learning models that generalise the data because a form of *regularisation* has been applied while learning (Geman et al., 1992). Regularisation reduces the variance, such that the spurious details are smoothed out, and the impact of noise is reduced.

3.1 The penalised (log-)likelihood approach

As we saw, the straight-forward approach to learning parameters via the likelihood $\Pr(\boldsymbol{d}|m,\boldsymbol{\theta})$, is to fix the model, and then use the ML estimates to actually fit the model. However, using the same approach for learning the model, it is well-known that this will result in *overfitting*. The learned model will have no (or very few) conditional independence restrictions.

In order to counteract overfitting, a measure of model complexity is introduced. The number of *free parameters*:

$$\kappa(m) = \sum_{i=1}^{p} |\Omega_{X_i} - 1| \cdot |\Omega_{\boldsymbol{X}_{pa(i)}}|$$

is the number of parameters that need to be estimated for a given model; for every parent set configuration, all configurations of the child need to be determined (except one, because the probability sums to unity). Hence, for dense DAG models, there are more free parameters. The model selected is then:

$$\hat{m} = \arg\max_{m} \log \Pr(\boldsymbol{d}|m,\hat{\boldsymbol{\theta}}) + f(m)$$

Here $f(m)$ is a *penalty term* that is responsible for regularisation, and $\hat{\boldsymbol{\theta}}$ is the ML estimate of $\boldsymbol{\theta}$.

If $f(m) = -\kappa(m)$ then the AIC criterion (Akaike, 1974) is obtained, derived from decision theory. Setting $f(m) = -\frac{\kappa(m)}{2}\log c$ we get the BIC criterion (Schwarz, 1978), based on the large sample approximation from Bayesian theory, or equivalently, the (negative) MDL criterion based on coding theory (Grünwald et al., 2005). We refer to Bouckaert, 1995 for a more thorough treatment of these metrics for learning Bayesian network models.

For all scores hold that they decompose, i.e., they are a sum (or product prior to taking the logarithm) of penalised terms, one term for each vertex. This is easy to see, since both the logarithm of the product likelihood in eq. 3.1 and $f(m)$ consists of a sum for $i = 1, \ldots, p$, i.e., one term per vertex.

3.2 The Bayesian approach

A Bayesian does not approach model learning directly from a penalised likelihood point of view. Moreover, model selection is not even in line with the Bayesian paradigm. In the same way that we were interested in the posterior parameter distribution when learning the parameter, the entire posterior model distribution $\Pr(M|\boldsymbol{d})$ is of interest when learning models. In Chapter 4, Section 3 we return to this issue.

If a Bayesian has to do model selection, usually due to computational considerations, it is usually determined as the MAP (maximum a posteriori) model from the posterior model distribution. Hence, the model selected is merely a summary statistic of the entire posterior model distribution.

As was the case with parameter learning, the posterior model distribution is obtained via Bayes' law:

$$\Pr(M|\boldsymbol{d}) = \frac{\Pr(\boldsymbol{d}|M) \cdot \Pr(M)}{\sum_m \Pr(\boldsymbol{d}|m) \cdot \Pr(m)} \propto \Pr(\boldsymbol{d}|M) \cdot \Pr(M) \tag{3.12}$$

The MAP model is the model for which $\Pr(M|\boldsymbol{d})$ is maximal. In Bayes' law, we may disregard the normalising term because it is constant for a fixed data set. It therefore suffices to maximise the numerator $\Pr(\boldsymbol{d}, M)$ with respect to M. The distribution $\Pr(M)$ can model any constraint one may have on the independences, directions or otherwise. Angelopoulos and Cussens, 2001 discuss how this can be done. In the absence of prior knowledge, a uniform prior can be used, i.e., all models are assigned the same prior probability $1/|\Omega_M|$. In that case we only need to maximise $\Pr(\boldsymbol{d}|M)$ because the prior is constant.

3.2.1 The marginal likelihood

The term $\Pr(\boldsymbol{d}|M)$ is the *marginal likelihood* not to be confused with the likelihood function $\Pr(\boldsymbol{d}|M, \boldsymbol{\Theta})$. In the marginal likelihood the parameter is integrated out with respect to the prior parameter distribution:

$$\Pr(\boldsymbol{d}|M) = \int \Pr(\boldsymbol{d}|M, \boldsymbol{\theta}) \cdot \Pr(\boldsymbol{\theta}|M) d\boldsymbol{\theta} \tag{3.13}$$

Hence, the marginal likelihood is the expectation of the likelihood with respect to the prior. Note that the marginal likelihood coincides with the normalising term in eq. 3.5. By rearranging and isolating the denominator we get:

$$\begin{aligned}
\Pr(\boldsymbol{d}|M) &= \frac{\Pr(\boldsymbol{d}|M,\boldsymbol{\Theta})\cdot\Pr(\boldsymbol{\Theta}|M)}{\Pr(\boldsymbol{\Theta}|\boldsymbol{d},M)} \\
&= \frac{\prod_{i=1}^{p}\prod_{\boldsymbol{x}_{pa(i)}}\prod_{x_i}\Theta_{x_i|\boldsymbol{x}_{pa(i)}}^{n(x_i,\boldsymbol{x}_{pa(i)})+\alpha(x_i,\boldsymbol{x}_{pa(i)})-1}\cdot Z(\boldsymbol{x}_{pa(i)},\boldsymbol{\alpha})}{\prod_{i=1}^{p}\prod_{\boldsymbol{x}_{pa(i)}}\prod_{x_i}\Theta_{x_i|\boldsymbol{x}_{pa(i)}}^{n(x_i,\boldsymbol{x}_{pa(i)})+\alpha(x_i,\boldsymbol{x}_{pa(i)})-1}\cdot Z(\boldsymbol{x}_{pa(i)},\boldsymbol{\alpha}+\boldsymbol{n})} \\
&= \prod_{i=1}^{p}\prod_{\boldsymbol{x}_{pa(i)}}\prod_{x_i}\frac{Z(\boldsymbol{x}_{pa(i)},\boldsymbol{\alpha})}{Z(\boldsymbol{x}_{pa(i)},\boldsymbol{\alpha}+\boldsymbol{n})} \qquad (3.14)
\end{aligned}$$

The equation reduces to the product of the ratios of the normalising factors of the prior Dirichlet and the posterior Dirichlet. Using eq. 3.7 we write eq. 3.14:

$$\prod_{i=1}^{p}\prod_{\boldsymbol{x}_{pa(i)}}\frac{\Gamma\Big(\alpha(\boldsymbol{x}_{pa(i)})\Big)}{\Gamma\Big(\alpha(\boldsymbol{x}_{pa(i)})+n(\boldsymbol{x}_{pa(i)})\Big)}\prod_{x_i}\frac{\Gamma\Big(\alpha(x_i,\boldsymbol{x}_{pa(i)})+n(x_i,\boldsymbol{x}_{pa(i)})\Big)}{\Gamma\Big(\alpha(x_i,\boldsymbol{x}_{pa(i)})\Big)} \qquad (3.15)$$

This gives the probability of the data under model M. The formula was derived from a somewhat different point of view in Cooper and Herskovits, 1992; Heckerman et al., 1995 where the marginal likelihood scoring metric was referred to as the *Bayesian Dirichlet scoring metric* (BD-metric).

The marginal likelihood in eq. 3.13 is quite different from the penalised likelihood scoring approach, where the parameter isn't integrated out, but is taken to be the ML estimate. The marginal likelihood takes the entire range of possible parameter assignments into consideration by explicitly weighing according to the parameter distribution. In contrast to the penalised likelihood approach, there is no explicit penalty term in the Bayesian approach; implicitly overfitting is still taken care of. In Section 3.2.2 and Section 3.3 we return to the issue, but here present the intuitive explanation: Recall from the discussion about the number of free parameters, that for dense DAG models, more parameters need to be determined than for less dense DAG models. This means that $\Omega_{\boldsymbol{\Theta}}$

is large, i.e., the more parameters, the smaller a density each "point" in $\Omega_{\boldsymbol{\Theta}}$ is assigned by the prior distribution $\Pr(\boldsymbol{\Theta}|M)$. The expectation of the likelihood as given in eq. 3.13, means that each parameter "point" is plugged into the likelihood and is weighted with a relatively small value for large $\Omega_{\boldsymbol{\Theta}}$, hence the overall expectation is lower than for small $\Omega_{\boldsymbol{\Theta}}$.

Another way of interpreting the marginal likelihood is via the so-called predictive-sequential approach called *prequential*; see for instance Seillier-Moiseiwitsch et al., 1992; Dawid, 1984. From the iid assumption, it follows that the marginal likelihood can be written as a product of conditional probabilities:

$$\begin{aligned}
\Pr(\boldsymbol{d}|M) &= \prod_{j=1}^{c} \Pr(\boldsymbol{d}^j|\boldsymbol{d}^1,\ldots,\boldsymbol{d}^{j-1},M) && (3.16)\\
&= \prod_{j=1}^{c} \Pr(\boldsymbol{d}^j|\boldsymbol{\theta}=\mathrm{E}[\boldsymbol{\Theta}|\boldsymbol{d}^1,\ldots,\boldsymbol{d}^{j-1}],M)\\
&= \prod_{j=1}^{c}\prod_{i=1}^{p} \frac{n^j(x_i^j,\boldsymbol{x}_{pa(i)}^j)+\alpha(x_i^j,\boldsymbol{x}_{pa(i)}^j)}{n^j(\boldsymbol{x}_{pa(i)}^j)+\alpha(\boldsymbol{x}_{pa(i)}^j)} && (3.17)
\end{aligned}$$

where $n^j(\cdot)$ is the function returning the counts from $\boldsymbol{d}$ until record j. In terms of $n(\cdot)$, it corresponds to the counts extracted from the contingency tables based on the data sample $(\boldsymbol{d}^1,\ldots,\boldsymbol{d}^j)$.

In eq. 3.16, each term is the probability of an instantiation of $\boldsymbol{X}$ conditional on previous observations. From eq. 2.5 we know that each term reduces to a product of expectations of the parameters with respect to the posterior Dirichlet, where the posterior is conditional on all records "seen thus far". Hence, by plugging in the expectation of the Dirichlet given in eq. 3.10, eq. 3.17 is obtained. Notice that the decomposition given in eq. 3.16 resembles a form of sequential *cross-validation*, i.e., use what has been seen thus far to test how well a particular model can predict the next observation.

The marginal likelihood admits to a recursive factorisation according to M in the same way that the BN does, viz. eq. 2.1:

$$\Pr(\boldsymbol{d}|M)=\prod_{i=1}^{p}\prod_{\boldsymbol{x}_{pa(i)}}\prod_{x_i}\frac{Z(\boldsymbol{x}_{pa(i)},\boldsymbol{\alpha})}{Z(\boldsymbol{x}_{pa(i)},\boldsymbol{\alpha}+\boldsymbol{n})}=\prod_{i=1}^{p}\Pr(\boldsymbol{d}_i|\boldsymbol{d}_{pa(i)},M) \quad (3.18)$$

where $\boldsymbol{d}_i$ denotes the column of $\boldsymbol{d}$ referring to X_i. Each term in this product is a predictive measure: given the c realisations (from the data

sample) for the parent variables, what is the probability of the c realisations for the child variable? Model selection says that we should per variable choose the parent set for which this prediction is best, because this is the best model from a prediction point of view.

The decomposition in eq. 2.1 is perhaps not surprising because the Dirichlet distribution is defined for each multinomial contingency table (due to local and global parameter independence), yet it is an important property when actually learning models. Recall that the penalised likelihood scoring criteria in Section 3.1 also are decomposable.

The marginal likelihood plays a significant role, not only when learning models, but as we will see in Chapter 5, also when dealing with incomplete data.

3.2.2 Determining the hyper parameter

The product Dirichlet is parameterised by the hyper parameter $\boldsymbol{\alpha}$. The hyper parameter captures the prior belief, before observing data. In the context of model learning, it turns out that the hyper parameter primarily is responsible for the degree of regularisation imposed.

For learning parameters, the hyper parameter can be specified in advance *given a particular DAG model*. For learning models rather than parameters, we would have to specify the hyper parameter for every possible DAG model. This is obviously infeasible to do by a human expert, for one because of the large number of different models, and also because from a domain perspective many models make no sense to the expert. Rather we would like to derive priors "automatically" for an arbitrary DAG model given that we have specified a probable DAG model m' and corresponding parameter $\boldsymbol{\theta}^{m'}$ (thus a full BN) that we think captures the prior *quantitative* knowledge.

Along with such a reference BN, an *equivalent sample size* (ESS) should be given, then $\alpha(x_i, \boldsymbol{x}_{pa(i)})$ is defined by (with $pa(i)$ being the indices of the parents in m):

$$\alpha(x_i, \boldsymbol{x}_{pa(i)}) = ESS \cdot \Pr(x_i, \boldsymbol{x}_{pa(i)} | m', \boldsymbol{\theta}^{m'}) \tag{3.19}$$

As the name suggests, the ESS is the number of prior observations on which the prior knowledge of the expert is based, i.e., what is the equivalent to the prior knowledge in terms of prior observations? From an intuitive point of view this perhaps makes sense, but formalising this relationship or mapping is pretty much impossible, and therefore

remains rather vague. The name *equivalent sample size* and the corresponding interpretation is however rather deceptive, because in practice the ESS is mainly responsible for the degree of regularisation imposed when learning models from data (Steck and Jaakkola, 2002).

PROPOSITION 3.1 *The ESS determines the overall degree of regularisation imposed when learning, and is distributed via* $\Pr(\boldsymbol{X}|m', \boldsymbol{\theta}^{m'})$ *over different parts of* M.

To see why this is, note that the prior distribution $\Pr(\boldsymbol{\Theta}|M) \equiv \Pr(\boldsymbol{\Theta}|M, \boldsymbol{\alpha})$ may be regarded as a function of $\boldsymbol{\alpha}$. In section 3.2.1 we already established that when the size of $\Omega_{\boldsymbol{\Theta}}$ increases, $\Pr(\boldsymbol{\Theta}|M)$ assigns a small density to each "point" in $\Omega_{\boldsymbol{\Theta}}$. However, for decreasing $\boldsymbol{\alpha}$, $\Pr(\boldsymbol{\Theta}|M)$ will also assign a smaller density to each "point" in $\Omega_{\boldsymbol{\Theta}}$, since the normalising factor given in eq. 3.7 decreases when $\boldsymbol{\alpha}$ goes to zero.

The ESS is distributed over all configurations $x_i, \boldsymbol{x}_{pa(i)}$ via eq. 3.19. For a complex DAG model M there are many configurations (large parent sets), and each $\alpha(x_i, \boldsymbol{x}_{pa(i)})$ is small because they share the ESS. This means that the normalising factor will be small, hence less complex DAG models are preferred. Thus, a large ESS implies weak regularisation, and a small ESS implies strong regularisation.

From the above discussion it follows that $\Pr(\boldsymbol{X}|m', \boldsymbol{\theta}^{m'})$ really distributes the "amount of regularisation" over different parts of M, i.e., the degree of regularisation for the vertices of M. In this respect it may be very difficult to specify such a BN in advance (even though only a single BN needs to be specified), because the notion of "distributing the regularisation" is very vague. In particular if we expect an expert to be able to specify such a BN, she will probably not be able to do so let alone grasp the very notion of regularisation.

In the literature it has been proposed to choose the prior hyper parameters according to the following metrics and methods:

- The *Bayesian Dirichlet equivalent* (BDe) is the method just described, where an ESS is chosen, and a distribution is defined that assigns $\alpha(x_i, \boldsymbol{x}_{pa(i)})$ to each Dirichlet.

  The BDe-metric is *score equivalent* which means that equivalent DAG models have the same marginal likelihood (Heckerman et al., 1995; Buntine, 1991); the reason is that the prior counts are "consistent" because they are linked to each other through the distribution defined in terms of the BN, $(m', \boldsymbol{\theta}^{m'})$.

- The *Bayesian Dirichlet equivalent uniform* (BDeu) is the same as the BDe-metric, but m' is the empty graph, and $\boldsymbol{\theta}^{m'}$ consists of parameters:

$$\Pr(X_i|m', \boldsymbol{\theta}^{m'}) = \frac{1}{|\Omega_{X_i}|}$$

  Consequently the joint over child and parents amounts to:

$$\Pr(X_i, \boldsymbol{X}_{pa(i)}|m', \boldsymbol{\theta}^{m'}) = \frac{1}{|\Omega_{X_i}|} \prod_{X_j \in \boldsymbol{X}_{pa(i)}} \frac{1}{|\Omega_{X_j}|}$$

  This metric is used when no prior information is available, or one is unable to specify in any detail the BN $(m', \boldsymbol{\theta}^{m'})$; only the ESS needs to be chosen to "tune" the overall degree of regularisation. The BDeu yields a uniform distribution of the ESS over $X_i, \boldsymbol{X}_{pa(i)}$.

  Because the BDe-metric is score equivalent, so is the BDeu-metric.

- The K2-metric simply says that all $\alpha(x_i, \boldsymbol{x}_{pa(i)}) = 1$.

  The K2-metric (Cooper and Herskovits, 1992) is not score equivalent, i.e., models that are equivalent do not have the same marginal likelihood; the prior counts are in no way related to each other. In general this is an undesirable property when learning models.

The K2-metric was originally used for learning with the K2-algorithm, presented in Cooper and Herskovits, 1992. This algorithm assumes that an ordering of the vertices is given a priori, and therefore score equivalence was not crucial. The BDeu-metric with an ESS of 1 is probably the most widely used metric in learning algorithms that are based on the marginal likelihood scoring criterion.

3.3 Marginal and penalised likelihood

The marginal likelihood is equivalent to the BIC/MDL penalised likelihood score, for an unlimited amount of data (Chickering and Heckerman, 1997; Bouckaert, 1995). For a finite amount of data, this does not hold, but for model selection, it is still possible to cast the Bayesian marginal likelihood in a functional form similar to the penalised likelihood scoring criteria, i.e., as the log-likelihood plus a penalty term. The functional similarity between all scores means that they can be used interchangeably in different theoretical contexts; the main difference lies in the regularisation used.

The joint distribution over BNs may be written as:

$$\Pr(M, \boldsymbol{\Theta}|\boldsymbol{d}) = \Pr(\boldsymbol{\Theta}|\boldsymbol{d}, M) \Pr(M|\boldsymbol{d}) \tag{3.20}$$

When learning BNs, we seek the MAP of that joint distribution, i.e., the most probable BN given data. The decomposition in 3.20 means that we may first select the model M for which $\Pr(M|\boldsymbol{d}) \propto \Pr(\boldsymbol{d}|M) \Pr(M)$ is maximal using the marginal likelihood, after which we may compute the MAP of $\Pr(\boldsymbol{\Theta}|\boldsymbol{d}, M)$ conditional on the model just selected.

THEOREM 3.1 *For model selection, the marginal likelihood can be transformed into an expression that has the functional form of a penalised (log-)likelihood.*

To obtain the MAP, we may reduce eq. 3.20 in the following way:

$$\begin{aligned}
\Pr(M, \boldsymbol{\Theta}|\boldsymbol{d}) &\propto \Pr(\boldsymbol{\Theta}|\boldsymbol{d}, M) \Pr(\boldsymbol{d}|M) \Pr(M) \\
&= \Pr(\boldsymbol{\Theta}|\boldsymbol{d}, M) \frac{\Pr(\boldsymbol{d}|M, \boldsymbol{\Theta}) \Pr(\boldsymbol{\Theta}|M)}{\Pr(\boldsymbol{\Theta}|\boldsymbol{d}, M)} \Pr(M) \\
&= \Pr(\boldsymbol{d}|M, \boldsymbol{\Theta}) \Pr(\boldsymbol{\Theta}|M) \Pr(M)
\end{aligned}$$

By filling in $\Pr(\boldsymbol{d}|M, \boldsymbol{\Theta})$ and $\Pr(\boldsymbol{\Theta}|M)$ and slightly rewriting we get:

$$\begin{aligned}
&\prod_{i=1}^{p} \prod_{\boldsymbol{x}_{pa(i)}} \prod_{x_i} \Theta_{x_i|\boldsymbol{x}_{pa(i)}}^{n(x_i, \boldsymbol{x}_{pa(i)})} \prod_{i=1}^{p} \prod_{\boldsymbol{x}_{pa(i)}} \prod_{x_i} \Theta_{x_i|\boldsymbol{x}_{pa(i)}}^{\alpha(x_i, \boldsymbol{x}_{pa(i)})-1} Z(\boldsymbol{x}_{pa(i)}, \boldsymbol{\alpha}) \Pr(M) \\
&= \underbrace{\prod_{i=1}^{p} \prod_{\boldsymbol{x}_{pa(i)}} \prod_{x_i} \Theta_{x_i|\boldsymbol{x}_{pa(i)}}^{n(x_i, \boldsymbol{x}_{pa(i)})+\alpha(x_i, \boldsymbol{x}_{pa(i)})-1}}_{\text{likelihood}} \underbrace{\prod_{i=1}^{p} \prod_{\boldsymbol{x}_{pa(i)}} Z(\boldsymbol{x}_{pa(i)}, \boldsymbol{\alpha}) \Pr(M)}_{\text{penalty}}
\end{aligned}$$

The likelihood is of the data *plus* the prior observations (minus 1), i.e., $\Pr(\boldsymbol{d}, \boldsymbol{\alpha} - \mathbf{1}|M, \boldsymbol{\Theta})$. It is worth observing that the penalty term is a combination of the prior model distribution $\Pr(M)$ *and* the normalising factor of the prior product Dirichlet distribution, which in turn is also dependent on M; we reduce this to the penalty term $f(M, \boldsymbol{\alpha})$. If we leave $\Pr(M)$ out of the equation, we have that similar to BIC/MDL and the AIC scores, the rewrite of the marginal likelihood remains decomposable on a per vertex basis. If $\Pr(M)$ is included, then this prior needs to decompose as well.

We know the closed form solution for the parameter that maximises the likelihood based on $(\boldsymbol{d}, \boldsymbol{\alpha}-\mathbf{1})$, namely the ML estimator $\hat{\boldsymbol{\theta}}$ (equivalent

to the MAP given $\boldsymbol{d}$), and since the penalty term does not depend on the choice of parameters, this solution also maximises $\Pr(M, \boldsymbol{\Theta}|\boldsymbol{d})$. Hence, we have that:

$$\hat{m} = \arg\max_m \log \Pr(\boldsymbol{d}, \boldsymbol{\alpha} - \mathbf{1}|m, \hat{\boldsymbol{\theta}}) + \log f(m, \boldsymbol{\alpha}) = \arg\max_m \log \Pr(m|\boldsymbol{d})$$

Note that for a fixed model, increasing hyper parameters increases the score. This is in line with the comments made about regularisation in Sections 3.2.1 and 3.2.2.

It is interesting to observe that the role of the prior hyper parameters in fact is twofold: they act as prior counts by adding to $\boldsymbol{d}$ the number of instances observed "in the past", *and* they determine the degree of regularisation.

For large samples, we have for the likelihood term that $\Pr(\boldsymbol{d}, \boldsymbol{\alpha} - \mathbf{1}|M, \boldsymbol{\Theta}) \approx \Pr(\boldsymbol{d}|M, \boldsymbol{\Theta})$, since $\boldsymbol{\alpha}$ will have an insignificant influence. This means that the main role of $\boldsymbol{\alpha}$ is to determine the degree of regularisation (again leaving $\Pr(M)$ out of the picture). That the prior counts have this effect is rather surprising, especially when compared to Bayesian parameter learning. For parameter learning, $\boldsymbol{\alpha}$ is treated in the same way as the sufficient statistics from $\boldsymbol{d}$; they are simply added together, and they are essentially treated as a batch. For model learning on the other hand, this "adding together" plays only a minor role.

3.4 Search methodologies

For parameter learning, both the Maximum-Likelihood approach as well as the Bayesian approach provide optimal solutions that can be found as closed functions of the sufficient statistics; no search procedure needs to be employed in order to establish the parameter estimates. For model selection, things are quite different.

Given a suitable *search space*, a *search strategy* and and some *scoring metric*, we are able to implement methods that perform model selection. Hence, the "optimal" model is searched for by traversing the search space, and the scoring metric (which to choose depends on the desired type and degree of regularisation) is the measure that determines the degree of optimality. It is the search strategy that specifies how the space is traversed.

3.4.1 Space

The search space defines the space of models that encode conditional independence statements. As previously discussed in chapter 3, Section

3.2, several DAGs encode the same conditional independence assumptions, meaning that the scoring metrics (except the marginal likelihood together with the K2-metric) for these DAGs are the same. This suggests that a search space with DAGs is larger than strictly necessary. The canonical representation of essential graphs would form a more suitable search space. However, the scoring metrics are defined on DAG models, and the computational burden of converting between DAG and essential graphs can be substantial; we note that recently (Chickering, 2002a) improvements have been made in this direction.

In Castelo and Kocka, 2003; Chickering, 2002a the impact of choosing the essential graph space instead of the DAG space is discussed. In that respect, the ratio of number of DAGs over the number of essential graphs is important, and is reported to be about 4 (Gillispie and Perlman, 2001; Kocka, 2001). This means that the average number of equivalent DAG models per essential graph is about 4, i.e., for a set of conditional independence statements encoded via an essential graph, there are on average 4 DAG models that can encode the same set of conditional independences. Moreover, for dense essential graphs, there are more equivalent DAGs, but the number of equivalent DAGs decreases rapidly as the essential graphs become less dense.

The covered arc reversal discussed in Chapter 2, section 3.2 provides a mechanism for moving between equivalent DAG models; by performing covered arc reversals, all (on average) 4 equivalent DAGs are visited. Hence a DAG search space can from an operational point of view simulate an essential graph space via a sub-process performing repeated covered arc reversals as the search strategy is employed. The search space remains the space of DAGs, but via the repeated covered arc reversals the search space effectively resembles the smaller essential graph space.

The covered arc reversal process is implemented in a non-deterministic way: For every DAG model, a covered arc is picked at random, and is reversed. After the reversal an equivalent DAG model is obtained, and once again a covered arc is reversed, etc. After a number of covered arc reversals, all DAGs in the equivalence class have been visited. Because the average number of equivalent DAGs is about 4, the number of covered arc reversals may be kept relatively small. For dense DAG models a larger number of reversals is preferred because more equivalent DAGs exist.

3.4.2 Strategy

The search strategy is intimately linked to the basic transformational operations one can perform on a DAG with limited computational costs, including the costs of computing the score after such a transformation. From a computational point of view it is advantageous to move around the DAG space in small steps. This corresponds to only modifying the current DAG model (corresponding to some "location" in the search space), by an arc transformation between two vertices X_i and X_j:

Arc addition: Change $X_i \not\rightarrow X_j$ to $X_i \rightarrow X_j$.

Arc removal: Change $X_i \rightarrow X_j$ to $X_i \not\rightarrow X_j$.

Arc reversal: Change $X_i \rightarrow X_j$ to $X_i \leftarrow X_j$.

Of these 3 operations, arc addition and arc reversal are non-local because they may introduce a cycle. Hence, more vertices need to be considered than X_i and X_j alone to check if these transformations are legal. A simple depth-first search reveals if a cycle is introduced.

From a scoring perspective, all 3 arc operations are local to X_i and X_j, because the metrics enjoy the property of decomposition. For two DAG models, only those terms in the marginal likelihood that pertain to vertices that have different parent sets are different. Hence, adding (removing) an arc to X_j means that all terms in the marginal likelihood remain unchanged, except the term pertaining to X_j, which has to be re-computed because its parent set changes. For reversal the two terms pertaining to X_i and X_j need to be re-computed.

Different learning algorithms have been proposed for learning DAG models, both deterministic and non-deterministic methods, see for instance Pelikan et al., 1999; Larrañaga et al., 1996; Heckerman, 1998; Buntine, 1991. Although the 3 elementary arc operations computationally provide an efficient way of moving around in the search space, these basic operations may not be the best or most logical choice for a given learning algorithm. Moreover, the elementary arc operations do not necessarily provide the best decomposition of the DAG model for reaching the highest scoring model, especially if the search space is severely spiked or otherwise non-smooth. In order to avoid getting stuck in local maxima, the 3 basic operations may not be "aggressive" enough to leave such an area.

A learning method that theoretically speaking only requires arc addition and arc removal to do optimal model selection, is based on the

so-called *inclusion-driven* approach, in which the search space is traversed in a way that respects the *inclusion boundary*. The inclusion boundary of a model from the essential graph space, consists of all essential graphs that have one edge more or one edge less than the current model. The inclusion boundary translates to the DAG model space via the covered arc reversals simulating the essential graph space. Given an *infinite* amount of data, the scoring metrics will always increase if addition or removal of an arc of a DAG model is "the right thing to do" with respect to the underlying distribution (the score is *consistent*), but will remain unchanged for equivalent DAG models. This means that we are guaranteed to reach the best model, if at every stage we move (in terms of a single arc addition or arc removal), to a model that is better than the current model. Several model selection algorithms have been developed that respect the inclusion boundary, such as GES by Chickering, 2002a; Chickering, 2002b, KES by Nielsen et al., 2003 and HCMC by Kocka and Castelo, 2001; Castelo and Kocka, 2003.

Unfortunately, for a *finite* data sample the optimality is not guaranteed, and it is no longer entirely clear whether adhering to an inclusion-driven learning approach is a big advantage over other learning approaches that also employ covered arc reversal to simulate the space of essential graphs. It might very well be that the performance of existing algorithms, that do not explicitly obey the inclusion boundary, can be improved by searching in the simulated essential graph space instead of the DAG space. This may not necessarily be because covered arc reversals also yield a smaller search space compared to the DAG search space, but because the covered arc reversals in fact make the search space smoother when we restrict the search strategy to one that only employs the 3 elementary arc operations.

Chapter 4

MONTE CARLO METHODS AND MCMC SIMULATION

In this chapter we treat the theory required for learning about posterior distributions, and we lay the foundation needed in order to learn BNs from incomplete data. We also develop a Markov chain Monte Carlo sampler called MB-MCMC for obtaining DAG models from the posterior model distribution given complete data.

Monte Carlo methods is a broad topic, and in this chapter we only present the essentials. See for instance Robert and Casella, 2002; Gilks et al., 1996 for a thorough treatment of different Monte Carlo techniques, including Markov chain Monte Carlo simulation and practical applications.

1. Monte Carlo methods

We investigate the generic problem of computing the following summation (replace sums with integrals for continuous variables):

$$\mathrm{E}_{\Pr}[h(\boldsymbol{X})] = \sum_{\boldsymbol{x}} h(\boldsymbol{x}) \Pr(\boldsymbol{x}) \tag{4.1}$$

In particular, $h(\boldsymbol{X})$ can be a probability distribution, for instance the conditional distribution $\Pr(\boldsymbol{Y}|\boldsymbol{X})$, in which case the expectation coincides with the distribution $\Pr(\boldsymbol{Y})$.

If eq. 4.1 is difficult to solve analytically or is infeasible to compute because of an extremely large cardinality of $\Omega_{\boldsymbol{X}}$, it can be approximated. Sample $\boldsymbol{X}^{(1)}, \ldots, \boldsymbol{X}^{(n)}$ from $\Pr(\boldsymbol{X})$, i.e., $\boldsymbol{X}^{(t)} \sim \Pr(\boldsymbol{X})$, and compute

the *empirical average* of eq. 4.1:

$$\mathrm{E}_{\Pr}[h(\boldsymbol{X})] \approx \frac{1}{n}\sum_{t=1}^{n} h(\boldsymbol{x}^{(t)}) \tag{4.2}$$

The Law of Large Numbers guarantees that the approximation converges towards the expectation in eq. 4.1 as $n \to \infty$.

For one reason or another it might be the case that sampling from $\Pr(\boldsymbol{X})$ is undesirable. For instance, it may be computationally expensive to produce realisations, or the distribution may be such, that it can be evaluated only up to a normalising term, making it difficult or impossible to sample from. Many computational problems encountered in Bayesian statistics in fact boil down to not being able to determine the normalising term, as we will see in section 2. This means that even solving the approximation of eq. 4.1 in terms of eq. 4.2 may be problematic.

In order to tackle these problems, more sophisticated *Monte Carlo* techniques may help out. We start off by introducing another Monte Carlo method called importance sampling, and then discuss Markov chain Monte Carlo sampling.

1.1 Importance sampling

A slightly more advanced Monte Carlo method than the approximation in eq. 4.2 is the so-called *importance sampler*. Suppose that we don't sample from $\Pr(\boldsymbol{X})$, but instead we sample from another distribution $\Pr'(\boldsymbol{X})$, e.g., it may be computationally cheap to do so.

First off, rewrite the summation in eq. 4.1:

$$\begin{aligned}
\sum_{\boldsymbol{x}} h(\boldsymbol{x})\Pr(\boldsymbol{x}) &= \frac{1}{\sum_{\boldsymbol{x}} \Pr(\boldsymbol{x})} \sum_{\boldsymbol{x}} h(\boldsymbol{x})\Pr(\boldsymbol{x}) \\
&= \frac{1}{\sum_{\boldsymbol{x}} \frac{\Pr(\boldsymbol{x})}{\Pr'(\boldsymbol{x})}\Pr'(\boldsymbol{x})} \sum_{\boldsymbol{x}} h(\boldsymbol{x})\frac{\Pr(\boldsymbol{x})}{\Pr'(\boldsymbol{x})}\Pr'(\boldsymbol{x}) \\
&= \frac{1}{\mathrm{E}_{\Pr'}[\frac{\Pr(\boldsymbol{X})}{\Pr'(\boldsymbol{X})}]} \mathrm{E}_{\Pr'}[h(\boldsymbol{X})\frac{\Pr(\boldsymbol{X})}{\Pr'(\boldsymbol{X})}]
\end{aligned} \tag{4.3}$$

Sample $\boldsymbol{X}^{(1)}, \ldots, \boldsymbol{X}^{(n)}$ from $\Pr'(\boldsymbol{X})$, and the empirical average of eq. 4.3 becomes:

$$\frac{1}{\sum_{t=1}^{n} w_t} \cdot \sum_{t=1}^{n} w_t \cdot h(\boldsymbol{x}^{(t)}) \tag{4.4}$$

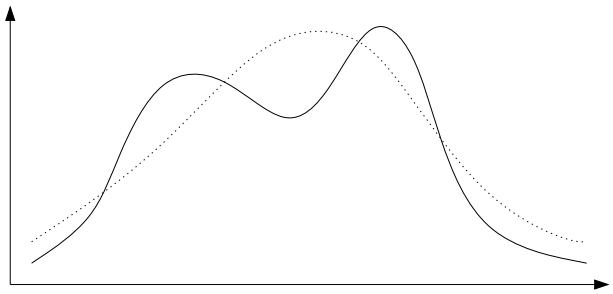

Figure 4.1. The dotted line is $\Pr'(\boldsymbol{X})$, and the solid line is $\Pr(\boldsymbol{X})$.

with weights w_t:

$$w_t = \frac{\Pr(\boldsymbol{x}^{(t)})}{\Pr'(\boldsymbol{x}^{(t)})} \tag{4.5}$$

When $\Pr'(\boldsymbol{X}) > 0$ whenever $\Pr(\boldsymbol{X}) > 0$, it holds that as n increases the approximation becomes more accurate. An essential observation is that the *constant* normalising term of $\Pr(\boldsymbol{X})$ cancels out because the weights as given in eq. 4.5 are normalised in eq. 4.4. This implies that we need only be able to evaluate $\Pr(\boldsymbol{X})$ up to this normalising constant. In fact, the normalising term of $\Pr'(\boldsymbol{X})$ is eliminated as well.

Theoretically speaking, importance sampling puts very little restriction on the choice of sampling distribution; in particular, any strictly positive sampling distribution can be used. When using a uniform sampling distribution, the denominator of w_t is the same for all weights t, and are eliminated by normalisation. Also note that when $\Pr'(\boldsymbol{X})$ and $\Pr(\boldsymbol{X})$ are proportional, the sampler reduces to the empirical average in eq. 4.2.

The weights w_t in eq. 4.5 are called *importance weights* and compensate for the difference between the real distribution and the sampling distribution. Intuitively, for the standard empirical average in eq. 4.2, all samples drawn from $\Pr(\boldsymbol{X})$ have the same weight in the approximation. For importance sampling, any sample from $\Pr'(\boldsymbol{X})$ is, compared to $\Pr(\boldsymbol{X})$, either "over-sampled" (too frequently sampled) and receives a weight less than 1, or "under-sampled" (too infrequently sampled) and receives a weight larger than 1. Figure 4.1 illustrates the principle behind importance sampling. The dotted line is the sampling distribution, and the solid line is from the target distribution, from which we can't sample. Samples drawn at locations where the dotted line lies above the

solid plot, will be drawn more often than necessary, and vice versa. To correct for that mismatch, the importance weights are required.

1.1.1 Choice of the sampling distribution

Although for large n the importance sampling approximation will be good, the sampling distribution has a major impact on the performance of importance sampling. In fact, choosing an inappropriate sampling distribution can have disastrous effects (see for instance Geweke, 1989). The rewrite of the expectation in terms of $\Pr'(\boldsymbol{X})$ results in the variance:

$$\begin{aligned}
\mathrm{Var}_{\Pr'}[h(\boldsymbol{X})\frac{\Pr(\boldsymbol{X})}{\Pr'(\boldsymbol{X})}] &= \mathrm{E}_{\Pr'}[h(\boldsymbol{X})^2\frac{\Pr(\boldsymbol{X})^2}{\Pr'(\boldsymbol{X})^2}] - \mathrm{E}_{\Pr'}[h(\boldsymbol{X})\frac{\Pr(\boldsymbol{X})}{\Pr'(\boldsymbol{X})}]^2 \\
&= \sum_{\boldsymbol{x}} h(\boldsymbol{x})^2\frac{\Pr(\boldsymbol{x})^2}{\Pr'(\boldsymbol{x})^2}\Pr'(\boldsymbol{x}) - \mathrm{E}_{\Pr}[h(\boldsymbol{X})]^2 \\
&= \sum_{\boldsymbol{x}} h(\boldsymbol{x})^2\frac{\Pr(\boldsymbol{x})}{\Pr'(\boldsymbol{x})}\Pr(\boldsymbol{x}) - \mathrm{E}_{\Pr}[h(\boldsymbol{X})]^2 \\
&= \mathrm{E}_{\Pr}[h(\boldsymbol{X})^2\frac{\Pr(\boldsymbol{X})}{\Pr'(\boldsymbol{X})}] - \mathrm{E}_{\Pr}[h(\boldsymbol{X})]^2 \\
&= \sum_{\boldsymbol{x}} h(\boldsymbol{x})^2\frac{\Pr(\boldsymbol{x})^2}{\Pr'(\boldsymbol{x})} - \Big(\sum_{\boldsymbol{x}} h(\boldsymbol{X})\Pr(\boldsymbol{X})\Big)^2 \quad (4.6)
\end{aligned}$$

The second term in eq. 4.6 is independent of $\Pr'(\boldsymbol{X})$, so our choice of $\Pr'(\cdot)$ only affects the first term. Assuming that we want to be able to use a wide range of functions $h(\boldsymbol{X})$ that we don't know a priori, we restrict attention to the effect that the ratio $\Pr(\boldsymbol{X})^2/\Pr'(\boldsymbol{X})$ has on the variance in the first term. When this fraction is unbounded, the variance for many functions is infinite. This leads to general instability and slows convergence. Notice that the ratio becomes extremely large in the tails when $\Pr(\boldsymbol{X})$ is larger than $\Pr'(\boldsymbol{X})$ in that region. A bounded ratio is the best choice, and in particular, in the tails $\Pr'(\boldsymbol{X})$ should dominate $\Pr(\boldsymbol{X})$.

Suppose a single $\boldsymbol{x}^{(t)}$ is drawn from $\Pr'(\boldsymbol{X})$ from an area of very low probability (density), and $\Pr(\boldsymbol{x}^{(t)}) \gg \Pr'(\boldsymbol{x}^{(t)})$. Such a sample can have a major impact on the empirical average via importance sampling. The sample is assigned far too much importance compared to the remaining samples because the ratio $\Pr(\boldsymbol{x}^{(t)})/\Pr'(\boldsymbol{x}^{(t)})$ is very large. Now suppose that $\Pr'(\boldsymbol{X})$ is a reasonable approximation of $\Pr(\boldsymbol{X})$ *almost everywhere* except in a few areas, where the importance weights are off-scale. Even though the majority of samples contribute to a reasonable approximation

of the expectation, as soon as a sample is obtained from "a bad area", the approximation seriously deteriorates because the importance weight is so much larger compared to the importance weights associated with the samples from the "good areas". In such a case, it may be better to discard such a sample entirely. This should be done with some caution though. Deletion will generally not introduce bias if the large weight is due to a very small denominator (compared to the denominator of the other weights). If it turns out that the deletion of a large-weight proposal results in a more sensible mass distribution over the remaining sample proposals, then this indeed does indicate that the sample just deleted was "an accidental outlier" and can be deleted without problem. On the other hand, if the numerator in a large-weight sample is large (compared to the numerator of the other weights), one may want to keep such a sample, since it then comes from a "region of relatively high impact" on the empirical approximation. In Hesterberg, 1995 it is suggested to use a mixture distribution as proposal distribution to overcome the problem pertaining to large importance weights. Each component of $\Pr'(\boldsymbol{X})$ is $\Pr'_i(\boldsymbol{X})$ with weight v_i. From an operational point of view, drawing $\boldsymbol{X}^{(t)}$ from $\Pr'(\boldsymbol{X})$ means that with probability v_i we draw from $\Pr'_i(\boldsymbol{X})$. In calculating the importance weights however, we use $\Pr'(\boldsymbol{x}^{(t)})$ rather than $\Pr'_i(\boldsymbol{x}^{(t)})$. This way we may define distributions $\Pr'_i(\boldsymbol{X})$ that cover different areas of $\Pr(\boldsymbol{X})$, such that the importance weights remain bounded.

A general recommendation is to monitor the running variance of the importance weights because it gives a good indication of the mutual differences between the importance weights of the proposals sampled from $\Pr'(\boldsymbol{X})$. When the variance suddenly jumps or is instable, one should at least investigate if the sampling distribution is appropriate, or if deletion of a few samples may correct the problem. Large variance may indicate that the performance of the sampler is poor and that the approximation will be sub-optimal.

2. Markov chain Monte Carlo—MCMC

Importance sampling offers a way of approximating $\mathrm{E}[h(\boldsymbol{X})]$ with respect to a distribution $\Pr(\boldsymbol{X})$ only known up to the normalising term, via some sampling distribution $\Pr'(\boldsymbol{X})$. Markov chain Monte Carlo (MCMC) is a whole range of methods that provide an alternative way of solving the generic problem of approximating the expectation in eq. 4.1.

Importance sampling is a so-called direct sampling method where at least partial knowledge of the distribution from which samples are re-

quired is needed to get a reasonable approximation. These methods are by nature non-iterative, and they do not adapt to the target distribution. MCMC on the other hand is a more flexible and iterative method for handling very awkward distributions. See Neal, 1993 and Andrieu et al., 2003 for a walkthrough of basic MCMC theory.

The problems that we want to solve are usually cast in a Bayesian framework, and we will therefore slightly reformulate the original problem. Consider Bayes' law:

$$\Pr(\boldsymbol{X}|\boldsymbol{Y}) = \frac{\Pr(\boldsymbol{Y}|\boldsymbol{X})\Pr(\boldsymbol{X})}{\sum_{\boldsymbol{x}} \Pr(\boldsymbol{Y}|\boldsymbol{x})\Pr(\boldsymbol{x})} \tag{4.7}$$

The denominator $\Pr(\boldsymbol{Y}) = \sum_{\boldsymbol{x}} \Pr(\boldsymbol{Y}|\boldsymbol{x})\Pr(\boldsymbol{x})$ is responsible for normalisation. Hence, if the numerator can be computed, then sampling from $\Pr(\boldsymbol{X}|\boldsymbol{Y})$, and for instance approximating $\mathrm{E}[h(\boldsymbol{X})]$ with respect to that distribution, can be solved via importance sampling or MCMC. In general $h(\boldsymbol{X})$ may be any function defined on $\Omega_{\boldsymbol{X}}$. Henceforth we leave the function $h(\boldsymbol{X})$ out of the picture, and address the main problem, namely sampling from $\Pr(\boldsymbol{X}|\boldsymbol{Y})$. Sometimes we skip the conditional, and the problem is thus how to sample from $\Pr(\boldsymbol{X})$, when this distribution is known up to a normalising term.

2.1 Markov chains

MCMC is based on the construction of a *Markov chain*, where the $\boldsymbol{X}^{(t)}$'s are produced sequentially, beginning from $\boldsymbol{X}^{(0)}$, such that $\boldsymbol{X}^{(t+1)}$ depends on $\boldsymbol{X}^{(t)}$ only, but is independent of $\boldsymbol{X}^{(t-1)}, \boldsymbol{X}^{(t-2)}, \ldots, \boldsymbol{X}^{(0)}$ (a so-called *one-step memory* sequence):

$$\boldsymbol{X}^{(t+1)} \perp\!\!\!\perp \boldsymbol{X}^{(0)}, \ldots, \boldsymbol{X}^{(t-1)} | \boldsymbol{X}^{(t)}$$

The chain is constructed via *transition probabilities* $T(\boldsymbol{X}^{(t+1)}|\boldsymbol{X}^{(t)})$ corresponding to a conditional distribution for $\boldsymbol{X}^{(t+1)}$ given $\boldsymbol{X}^{(t)}$, and an initial distribution $\Pr_0(\boldsymbol{X}^{(0)})$. The distribution $\Pr_{t+1}(\boldsymbol{X}^{(t+1)})$ is then defined in terms of $\boldsymbol{X}^{(t)}$ via the transition:

$$\Pr_{t+1}(\boldsymbol{X}^{(t+1)}) = \sum_{\boldsymbol{x}^{(t)}} T(\boldsymbol{X}^{(t+1)}|\boldsymbol{x}^{(t)}) \Pr_t(\boldsymbol{x}^{(t)})$$

When the transition probabilities as defined here do not depend on t, the Markov chain is called (time) *homogeneous*.

2.1.1 The invariant target distribution

An *invariant distribution* $\Pr_*(\boldsymbol{X})$ for a Markov chain, is one that persists forever once reached:

$$\Pr_*(\boldsymbol{X}) = \sum_{\boldsymbol{x}'} T(\boldsymbol{X}|\boldsymbol{x}')\Pr_*(\boldsymbol{x}') \tag{4.8}$$

The invariant distribution, is the *target distribution*, i.e., the distribution from which we want to draw samples. An invariant distribution exists if the Markov chain satisfies the *detailed balance* condition. Detailed balance states that the transition probability is symmetric between $\boldsymbol{X}$ and $\boldsymbol{X}'$, when they are sampled from a distribution $\Pr(\boldsymbol{X})$. Formally, it should hold that:

$$T(\boldsymbol{X}'|\boldsymbol{X})\Pr(\boldsymbol{X}) = T(\boldsymbol{X}|\boldsymbol{X}')\Pr(\boldsymbol{X}') \tag{4.9}$$

In fact, this means that $\Pr(\boldsymbol{X}) = \Pr_*(\boldsymbol{X})$, i.e., the distribution must necessarily be an invariant distribution viz. eq. 4.8:

$$\begin{aligned}\sum_{\boldsymbol{x}'} T(\boldsymbol{X}|\boldsymbol{x}')\Pr(\boldsymbol{x}') &= \sum_{\boldsymbol{x}'} T(\boldsymbol{x}'|\boldsymbol{X})\Pr(\boldsymbol{X}) \\ &= \Pr(\boldsymbol{X})\end{aligned}$$

We stress that detailed balance is a sufficient but not necessary condition for ensuring invariance.

2.1.2 Reaching the invariant distribution

To guarantee that the invariant distribution is reached, the Markov chain needs to be *irreducible* and *aperiodic* (Tierney, 1994). The chain is irreducible if starting from $\boldsymbol{X}^{(0)}$ every region of the space from which we need to sample can eventually be reached with a positive probability. In figure 4.2, left, this is illustrated. The outline of the whole figure illustrates the actual state space, and the gray area corresponds to the area explored by an irreducible Markov chain. No matter where $\boldsymbol{X}^{(0)}$ is located, all states *communicate* and can be reached; no region is isolated and the chain is free to move to other states via other intermediate states. To the right, the chain is *reducible*. Again, the whole figure illustrates the state space from which samples are required. The two gray areas are not connected, indicating that only the region from which the initial $\boldsymbol{X}^{(0)}$ is located is explored. The whole state space from which we wish to sample is not covered because there is no way to reach the other region.

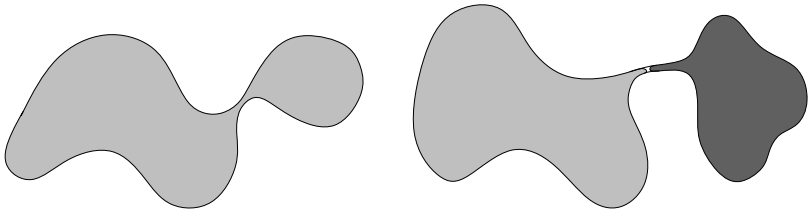

Figure 4.2. State spaces explored by the Markov chain. *Left:* irreducible. *Right:* reducible.

Aperiodicity guarantees that the chain does not get trapped in cycles, preventing the chain from getting the opportunity to reach other areas. It is quite easy to make sure that this does not happen; as long as there is a strictly positive probability of remaining in the current state, i.e., the probability of $\boldsymbol{X}^{(t)}$ and $\boldsymbol{X}^{(t+1)}$ having the same value is > 0, the chain will not get stuck in cycles.

To sum up, if a homogeneous Markov chain...

- has an *invariant* distribution, $\Pr(\boldsymbol{X})$,
- it is *irreducible*, and
- it is *aperiodic*,

the Markov chain will in the limit produce realisations from the invariant distribution, regardless of the starting point $\boldsymbol{X}^{(0)}$:

$$|\Pr_{t+1}(\boldsymbol{X}) - \Pr(\boldsymbol{X})| \to \mathbf{0} \text{ for } t \to \infty$$

Such a Markov chain is said to be *ergodic*.

In the next two sections we discuss two well-known MCMC methods that produce samples from some *desired* invariant distribution by building an ergodic Markov chain: the Metropolis-Hastings sampler, and the Gibbs sampler. Sometimes we say that a MCMC sampler is ergodic, in which case it refers to the Markov chain that is produced by the sampler.

2.1.3 Metropolis-Hastings sampling

Metropolis-Hastings MCMC (Metropolis et al., 1953; Hastings, 1970) by construction produces samples from the distribution, $\Pr(\boldsymbol{X})$, which is the invariant distribution for the Markov chain. Quite similar to importance sampling, a *proposal* distribution, $\Pr'(\boldsymbol{Y}|\boldsymbol{X})$, from which

we can sample exists. The Metropolis-Hastings algorithm produces a Markov chain $\boldsymbol{X}^{(0)}, \boldsymbol{X}^{(1)}, \ldots$ in the following way:

---

1 Draw $\boldsymbol{Y} \sim \Pr'(\boldsymbol{Y}|\boldsymbol{X}^{(t)})$

2 Take

$$\boldsymbol{X}^{(t+1)} = \begin{cases} \boldsymbol{y} & \text{with probability } \rho(\boldsymbol{x}^{(t)}, \boldsymbol{y}) \\ \boldsymbol{x}^{(t)} & \text{with probability } 1 - \rho(\boldsymbol{x}^{(t)}, \boldsymbol{y}) \end{cases}$$

where

$$\rho(\boldsymbol{X}, \boldsymbol{Y}) = \min\left\{ \frac{\Pr(\boldsymbol{Y}) \Pr'(\boldsymbol{X}|\boldsymbol{Y})}{\Pr(\boldsymbol{X}) \Pr'(\boldsymbol{Y}|\boldsymbol{X})}, 1 \right\}$$

The transition probability is the probability of proposing $\boldsymbol{Y}$ times the probability of accepting this candidate as $\boldsymbol{X}^{(t+1)}$; additionally, it includes the probability of proposing any state $\boldsymbol{X}'$ and rejecting it, remaining in the current state, $\boldsymbol{Y} = \boldsymbol{X}^{(t)}$. Thus the transition $T(\boldsymbol{Y}|\boldsymbol{X}^{(t)})$ becomes:

$$\rho(\boldsymbol{X}^{(t)}, \boldsymbol{Y}) \Pr'(\boldsymbol{Y}|\boldsymbol{X}^{(t)}) + I(\boldsymbol{Y} = \boldsymbol{X}^{(t)}) \sum_{\boldsymbol{x}'} (1 - \rho(\boldsymbol{X}^{(t)}, \boldsymbol{x}')) \Pr'(\boldsymbol{x}'|\boldsymbol{X}^{(t)})$$

where $I(\cdot)$ is the indicator function. We indeed have that $\Pr(\boldsymbol{X})$ is the invariant distribution, because detailed balance holds. For the second term this is easy to see. For the first term we distinguish cases. In the case $\frac{\Pr(\boldsymbol{Y}) \Pr'(\boldsymbol{X}|\boldsymbol{Y})}{\Pr(\boldsymbol{X}) \Pr'(\boldsymbol{Y}|\boldsymbol{X})} > 1$, we have that $\rho(\boldsymbol{X}, \boldsymbol{Y}) = 1$, and by applying eq. 4.9 it follows:

$$\begin{aligned} \Pr(\boldsymbol{X}) \rho(\boldsymbol{X}, \boldsymbol{Y}) \Pr'(\boldsymbol{Y}|\boldsymbol{X}) &= \Pr(\boldsymbol{X}) \Pr'(\boldsymbol{Y}|\boldsymbol{X}) \\ &= \Pr(\boldsymbol{X}) \Pr'(\boldsymbol{Y}|\boldsymbol{X}) \frac{\Pr(\boldsymbol{Y}) \Pr(\boldsymbol{X}|\boldsymbol{Y})}{\Pr(\boldsymbol{Y}) \Pr(\boldsymbol{X}|\boldsymbol{Y})} \\ &= \Pr(\boldsymbol{Y}) \Pr'(\boldsymbol{X}|\boldsymbol{Y}) \frac{\Pr(\boldsymbol{X}) \Pr(\boldsymbol{Y}|\boldsymbol{X})}{\Pr(\boldsymbol{Y}) \Pr(\boldsymbol{X}|\boldsymbol{Y})} \\ &= \Pr(\boldsymbol{Y}) \rho(\boldsymbol{Y}, \boldsymbol{X}) \Pr'(\boldsymbol{X}|\boldsymbol{Y}) \end{aligned}$$

In case $\frac{\Pr(\boldsymbol{Y}) \Pr'(\boldsymbol{X}|\boldsymbol{Y})}{\Pr(\boldsymbol{X}) \Pr'(\boldsymbol{Y}|\boldsymbol{X})} < 1$ we have $\rho(\boldsymbol{X}, \boldsymbol{Y}) = \frac{\Pr(\boldsymbol{Y}) \Pr'(\boldsymbol{X}|\boldsymbol{Y})}{\Pr(\boldsymbol{X}) \Pr'(\boldsymbol{Y}|\boldsymbol{X})}$ and $\rho(\boldsymbol{Y}, \boldsymbol{X}) = 1$, and it follows:

$$\Pr(\boldsymbol{X}) \rho(\boldsymbol{X}, \boldsymbol{Y}) \Pr'(\boldsymbol{Y}|\boldsymbol{X}) = \Pr(\boldsymbol{X}) \Pr'(\boldsymbol{Y}|\boldsymbol{X}) \frac{\Pr(\boldsymbol{Y}) \Pr'(\boldsymbol{X}|\boldsymbol{Y})}{\Pr(\boldsymbol{X}) \Pr'(\boldsymbol{Y}|\boldsymbol{X})}$$

$$\begin{aligned} &= \Pr(\boldsymbol{Y})\Pr'(\boldsymbol{X}|\boldsymbol{Y}) \\ &= \Pr(\boldsymbol{Y})\Pr'(\boldsymbol{X}|\boldsymbol{Y})\rho(\boldsymbol{Y},\boldsymbol{X}) \end{aligned}$$

Hence, the Markov chain has invariant distribution $\Pr(\boldsymbol{X})$.

The Metropolis-Hastings sampler depends on the ratios $\Pr(\boldsymbol{Y})/\Pr(\boldsymbol{X})$ and $\Pr'(\boldsymbol{X}|\boldsymbol{Y})/\Pr'(\boldsymbol{Y}|\boldsymbol{X})$, implying that the normalising term for both the proposal distribution and, more importantly, for $\Pr(\cdot)$ cancels out.

Irreducibility is guaranteed when $\Pr'(\boldsymbol{Y}|\boldsymbol{X})$ covers the entire sampling area on which the invariant distribution $\Pr(\boldsymbol{X})$ is defined, since then any point can be reached via a proposal. Aperiodicity is also guaranteed, since there is a non-zero probability of remaining in the current state. If the proposal distribution is strictly positive this means that the invariant distribution will be reached in the limit. Notice that this is quite similar to importance sampling, where a sufficient condition for convergence was that the sampling distribution was positive.

A special case of the Metropolis-Hastings MCMC occurs when we have that $\Pr'(\boldsymbol{Y}|\boldsymbol{X}) = \Pr'(\boldsymbol{Y})$, i.e., the proposal distribution is independent of the last state like in importance sampling. This leads to:

$$\rho(\boldsymbol{X},\boldsymbol{Y}) = \min\Big\{\frac{\Pr(\boldsymbol{Y})\Pr'(\boldsymbol{X})}{\Pr(\boldsymbol{X})\Pr'(\boldsymbol{Y})},1\Big\} = \min\Big\{\frac{\Pr(\boldsymbol{Y})/\Pr'(\boldsymbol{Y})}{\Pr(\boldsymbol{X})/\Pr'(\boldsymbol{X})},1\Big\} \quad (4.10)$$

The numerator and the denominator also arise in importance sampling as the importance weights. With importance sampling the samples are weighted in order to correct for the difference between $\Pr(\boldsymbol{X})$ and $\Pr'(\boldsymbol{X})$, whereas for Metropolis-Hastings MCMC, the unweighted samples can be used, as they provably do come from $\Pr(\boldsymbol{X})$ (at least in the limit), and the weights are not required in the direct sense. However, indirectly the weights are still needed for the ratio given in eq. 4.10.

2.1.4 Gibbs sampling

Gibbs sampling (Geman and Geman, 1984) can be seen as special case of Metropolis-Hastings sampling, where the transition probabilities are defined in terms of the conditionals of the invariant distribution. Often we are able to draw from these conditionals, without being able to draw from the joint invariant distribution.

The Gibbs sampler we present is defined in terms of *blocks* of variables. Each block is drawn conditional on the variables not part of the current block. Our presentation deviates from the customary treatment of Gibbs sampling. The reason is that the "normal" Gibbs sampler can

be regarded as a special case of block Gibbs sampling, and that the added value of blocking is beneficial in many respects (Liu, 1994; Liu et al., 1994). We illustrate this in Section 2.1.6, and in Section 3.3 we exploit blocking for learning DAG models.

By construction, the Gibbs sampler needs at least two blocks. We write $\boldsymbol{X} = \boldsymbol{Y} \cup \boldsymbol{Z}$, indicating that we partition $\boldsymbol{X}$ into two blocks: block one, $\boldsymbol{Y}$ and block two, $\boldsymbol{Z}$. For ease of exposition, we assume that the blocks are disjoint, but in general they do not need to be. The proposal only changes block $\boldsymbol{Y}$:

$$\boldsymbol{Y} \sim \Pr'(\boldsymbol{Y}|\boldsymbol{Z}) = \Pr(\boldsymbol{Y}|\boldsymbol{Z})$$

where the conditional side $\boldsymbol{Z}$ is unchanged between moves, and unconditional on the last state of block $\boldsymbol{Y}$. Again, this is similar to drawing from the sampling distribution in importance sampling.

The relationship between the proposal distribution and the invariant distribution is constant through the ratio:

$$\Pr(\boldsymbol{Z}) = \frac{\Pr(\boldsymbol{Y}, \boldsymbol{Z})}{\Pr(\boldsymbol{Y}|\boldsymbol{Z})} = \frac{\Pr(\boldsymbol{Y}, \boldsymbol{Z})}{\Pr'(\boldsymbol{Y}|\boldsymbol{Z})}$$

Similar to eq. 4.10, the acceptance ratio becomes:

$$\rho(\boldsymbol{U}, \boldsymbol{Y}) = \min\left\{\frac{\Pr(\boldsymbol{Y}, \boldsymbol{Z})/\Pr'(\boldsymbol{Y}|\boldsymbol{Z})}{\Pr(\boldsymbol{U}, \boldsymbol{Z})/\Pr'(\boldsymbol{U}|\boldsymbol{Z})}, 1\right\} = 1$$

yielding an acceptance rate of 1, meaning that all proposals are accepted.

Obviously only sampling $\boldsymbol{Y}$ means that the Markov chain can't be irreducible, since the proposal distribution only proposes changes to one block. By only sampling in this lower-dimensional space, it immediately follows that any point not in that dimension will remain fixed—it can't be reached. A minimal conditional for ensuring irreducibility is to propose changes to block $\boldsymbol{Z}$ as well, i.e., sample $\boldsymbol{Z}$ from the conditional. Formally speaking, this is achieved by combining several Gibbs samplers, one per block. This amounts to applying transitions in turn, one transition per block. The chain remains invariant because each separate block transition leaves the chain invariant. To see why this is, suppose that we start the sampler from the invariant distribution. Each block is now sampled from the conditional of the invariant distribution. This transition leaves the marginal distribution (that coincides with the marginal of the invariant distribution) of the other blocks intact. For the block that is sampled, the transition obviously also leaves the chain invariant.

This argument holds for all the blocks and therefore we may conclude that once the invariant distribution has been reached, a combination of block transitions leaves the chain invariant.

The Gibbs sampler is usually presented in the following way:

Assign $X_i \in \boldsymbol{X}$ to some block $\boldsymbol{B}_j, j = 1, \ldots, k$, such that X_i is *part of at least one block*. Let $\Pr(\boldsymbol{B}_j | \boldsymbol{B}_{r \neq j})$ be the conditional invariant distribution:

$$
\begin{aligned}
\boldsymbol{B}_1^{(t+1)} &\sim \Pr(\boldsymbol{B}_1 | \boldsymbol{b}_2^{(t)}, \ldots, \boldsymbol{b}_k^{(t)}) \\
\boldsymbol{B}_2^{(t+1)} &\sim \Pr(\boldsymbol{B}_2 | \boldsymbol{b}_1^{(t+1)}, \boldsymbol{b}_3^{(t)}, \ldots, \boldsymbol{b}_k^{(t)}) \\
&\vdots \\
\boldsymbol{B}_k^{(t+1)} &\sim \Pr(\boldsymbol{B}_2 | \boldsymbol{b}_1^{(t+1)}, \ldots, \boldsymbol{b}_{k-1}^{(t+1)}) \\
\boldsymbol{B}_1^{(t+2)} &\sim \Pr(\boldsymbol{B}_1 | \boldsymbol{b}_2^{(t+1)}, \ldots, \boldsymbol{b}_k^{(t+1)}) \\
&\vdots
\end{aligned}
$$

The realisations of $\boldsymbol{X}$ thus obtained, are coming from the invariant distribution, $\Pr(\boldsymbol{X})$. In particular if X_i is assigned to the singleton set $\boldsymbol{B}_i$ and $k = p$ (number of variables in $\boldsymbol{X}$), then the Gibbs sampler reduces to drawing from the so-called *full conditionals*; each draw is univariate conditional on $\boldsymbol{X} \setminus \{X_i\}$. This is also referred to as a *single-site Gibbs sampler*.

The visitation scheme as suggested above is not crucial for convergence. Random visitation, a systematic sweep or any other combination is possible. Depending on the problem at hand, one scheme may be better than the other. As long as all X_i of $\boldsymbol{X}$ are sampled "infinitely" often, the invariant distribution will be reached.

The Markov chain is also aperiodic, because there is a probability > 0 of remaining in the current state (of a particular block). All dimensions of the state space are considered by sampling from the corresponding conditional, providing a minimal condition for irreducibility. Together with the so-called *positivity requirement*, this provides a sufficient condition for irreducibility. The positivity requirement says that all the conditionals must be strictly positive. Hence, not only are all dimensions visited, but all values along those dimensions can be reached as well.

We illustrate an instance of Gibbs sampling in the context of sampling from a BN. Suppose we are given the BN $(m, \boldsymbol{\theta})$ representing the joint distribution $\Pr(\boldsymbol{X}|m, \boldsymbol{\theta})$, and that the distribution required is $\Pr(\boldsymbol{Z}|m, \boldsymbol{\theta})$ for only a subset of the variables, $\boldsymbol{Z} \subseteq \boldsymbol{X}$. Since Gibbs sampling returns realisations from $\Pr(\boldsymbol{X}|m, \boldsymbol{\theta})$, any marginal distribution of any subset can be estimated by way of counting the realisations. That is, estimate $\Pr(\boldsymbol{Z}|m, \boldsymbol{\theta})$ by using the empirical average of the realisations of $\boldsymbol{Z}$, i.e.:

$$\Pr(\boldsymbol{z}|m, \boldsymbol{\theta}) \approx \frac{1}{n} \sum_{t=1}^{n} I(\boldsymbol{z} \subseteq \boldsymbol{x}^{(t)})$$

By employing a univariate Gibbs sampler drawing from the full conditionals, the Markov blanket makes the sampling process easy. The full conditional distribution reduces to $\Pr(X_j|X_{j-1}, X_{j+1}, \ldots, X_p, m, \boldsymbol{\theta}) = \Pr(X_j|\boldsymbol{X}_{mb(j)}, m, \boldsymbol{\theta})$ because the Markov blanket shields off all influence from variables outside the Markov blanket. Following the BN decomposition, the univariate distribution is:

$$\Pr(X_j|\boldsymbol{X}_{mb(j)}, m, \boldsymbol{\theta}) = \frac{\theta_{X_j|\boldsymbol{X}_{pa(j)}} \prod_{i \in ch(j)} \theta_{X_i|\boldsymbol{X}_{pa(i)}}}{\sum_{x_j} \theta_{x_j|\boldsymbol{X}_{pa(j)}} \prod_{i \in ch(j)} \theta_{X_i|\boldsymbol{X}_{pa(i)}}}$$

from which each variable in $\boldsymbol{Z}$ is drawn in turn according to the normal Gibbs procedure. Notice, that any conditional distribution over a set of variables from the BN can be calculated similarly. In that case simply fix the variables on the conditional side (the so-called *evidence*), and proceed with the Gibbs sampling procedure by updating the other variables. In that respect, Gibbs sampling is a BN inference method, albeit an approximate one.

2.1.5 Mixing, burn-in and convergence of MCMC

From a computational point of view, the most important properties of MCMC samplers pertain to the following:

- the *mixing* of the chain,
- the *burn-in* time, and
- the *convergence*.

It is crucial to understand that MCMC produces *correlated* samples. In this regard it may be beneficial to see MCMC as a way of "walking

around" some state space, such that locations with high probability (density) are passed relatively often. At every step, the current location is returned, and this corresponds to a draw. MCMC is adaptive in the sense that it will have a tendency to seek areas of "mass" or "interest" rather than just walk around aimlessly.

Mixing refers to the long-term correlations between the states of the chain. It refers to how fast the states "forget" about the previous states, i.e., how far from an iid sample the state of the chain is. This captures a notion of how large the "steps" are when traversing the state space. In general we want consecutive realisations to be as close to iid as possible. Slow mixing implies long-term drifts or trends. The terms *mobility* or *acceleration* of a chain, refer to the mixing properties.

When starting an ergodic chain at time $t = 0$ with a realisation of $\boldsymbol{X}^{(0)}$ not sampled from the invariant distribution, the time it takes to reach the invariant distribution is referred to as the *burn-in*, i.e., the time it takes before samples can be regarded as coming from the target distribution. After the burn-in, we say that the chain has *converged*; the realisations from then on may be considered samples from the invariant distribution.

The question is how long the chain needs to be before the realisations are sufficiently close to the invariant distribution. Obviously, we would like to reach the invariant distribution with a minimum amount of computational effort usually meaning that we want to keep the "wasteful" burn-in realisations at a minimum. At the same time we also would like to be certain that the invariant distribution indeed *has* been reached. Prematurely assuming that samples come from the invariant distribution when in fact they are just from a burn-in is less than useful.

With poor mobility of the chain, the sampler explores the state space in an inefficient way, which in effect means that the burn-in period increases. Moreover, assuming that the chain indeed has converged, poor mixing means that more realisations are required in order to gather enough empirical information to get an impression of the invariant distribution. In that respect, the mixing properties of a chain is the most significant factor that determines the performance of MCMC.

Unless we can provide an initial $\boldsymbol{X}^{(0)}$ from the invariant distribution (no burn-in is required at all), diagnosing convergence of MCMC is necessary for a trustworthy result. Unfortunately this is a non-trivial problem. In practice diagnosing convergence using an automated procedure is quite difficult and problem dependent. This means that in general such automated methods are unsafe on their own. Many diag-

nostics are very technical, and for many problems they do not directly apply, or are difficult to implement; see for instance Gelman and Rubin, 1992.

Monitoring the realisations against t gives a reasonable impression of the range of values that are "possible" for the invariant distribution. When realisations are multivariate, scalar functions of the realisations may be monitored. A burn-in will lie outside this range and stand out in comparison to the remaining realisations. With poor mixing, a burn-in may be difficult to discern, since a certain trend may last for many iterations. We may be lead to think that the chain has reached the invariant distribution when it in fact has not. However, if we are able accelerate the chain, then we may monitor the realisations over a shorter time frame, and trends are easier to discern.

Admittedly, a graphical criterion for assessing convergence may be deceptive because it does obviously not say anything about "where the chain will be in a few moments from now". This is especially so if only a single chain is used. Running MCMC with different initial so-called overdispersed values with respect to the invariant distribution, and comparing the plots by visual inspection provides a better way of checking if the chain is stuck "by accident" or if all runs result in similar bands of realisations. Of course similar plots is no guarantee for convergence, but it indicates a certain robustness, because the different chains eventually all reach the same region meaning that minor non-deterministic disturbances have not resulted in "off-course" behaviour of the chains. However, running several parallel chains is from a computational point of view not the preferred choice for assessing convergence.

2.1.6 The importance of blocking

In the previous section the importance of the Markov chain mixing properly was stressed. In this section we address the issue of mixing in the context of the Gibbs sampler. Blocking plays an important role in this regard.

Suppose that we apply (block) Gibbs sampling to obtain samples from the invariant distribution $\Pr(\boldsymbol{Y}, \boldsymbol{Z})$, $\boldsymbol{X} = \boldsymbol{Y} \cup \boldsymbol{Z}$. We first sample from $\Pr(\boldsymbol{Y}|\boldsymbol{Z})$ and then from $\Pr(\boldsymbol{Z}|\boldsymbol{Y})$ etc. The values of the variables behind the conditional constrain the probability of sampling certain values from the distribution, e.g., when sampling from $\Pr(\boldsymbol{Z}|\boldsymbol{Y})$, the value that $\boldsymbol{Z}$ can take on, is constrained by the value of the last realisation of $\boldsymbol{Y}$. Although irreducibility is guaranteed when the conditionals are strictly positive, an unfortunate value on the conditional side may imply that

that the variance of the conditional distribution is very small. It may be highly improbable (though not impossible) to sample certain values "in the tail", effectively having a negative impact on the mobility of the chain. Theoretically the chain will reach the entire sampling space, in practice this may take very long, because the chain has maneuvered itself into a region that is comparable to a local optimum.

When dependence between $\boldsymbol{Z}$ and $\boldsymbol{Y}$ is strong, then the situation sketched may very well occur. If the dependence is absent, thus when $\Pr(\boldsymbol{Y}, \boldsymbol{Z}) = \Pr(\boldsymbol{Y}) \cdot \Pr(\boldsymbol{Z})$, we have that the conditionals reduce to $\Pr(\boldsymbol{Y})$ and $\Pr(\boldsymbol{Z})$, and the interaction effects are gone. For instance, when sampling from $\Pr(\boldsymbol{Z}|\boldsymbol{Y}) = \Pr(\boldsymbol{Z})$, a previous value assigned to $\boldsymbol{Y}$ does not influence the probability of drawing a particular value $\boldsymbol{z}$.

The above discussion suggests that we should look for blocks that are only weakly dependent since this prevents the chain from getting trapped (see also Roberts and Sahu, 1997). It may be difficult to devise such blocks where this is the case for all values one can assign to the blocks. In this respect a notion of *context* (in)dependence also plays a role. Two blocks may be strongly dependent for *some* values, yet for other values less. Hence, we seek blocks that are context independent on each other for *many* block assignments.

Additionally, Gibbs sampling allows for dynamic creation of blocks while running. The blocks need not *a priori* be defined before starting the Gibbs sampler, but the individual blocks may change. Also, the blocks need not be disjoint. We may profit from this dynamic way of defining blocks since it offers a large degree of freedom in order to accommodate the desire to sample blocks that don't constrain each other strongly.

A block we may regard as a set of variables that inherently belong together, or alternatively, as something that should be sampled together, because the individual variables strongly depend on each other, and splitting them has a negative impact on the mixing of the chain (Jensen et al., 1995). It may not always be easy to determine these blocks, but for some problems there is a natural decomposition which follows almost immediately from the problem description. Although a different branch of computer science, in evolutionary computation the importance of the right decomposition has also been acknowledged as being very important, and deceptive decompositions have been investigated (Pelikan et al., 2003). In many other ways there are actually striking similarities between Monte Carlo methods and the population based paradigm on which evolutionary computation is based. In Section 3.3 we will sam-

ple edges from the posterior model distribution, and we investigate how edges of a DAG model are best blocked to improve mixing.

As an example of blocking, suppose that samples are required from some distribution $\Pr(\boldsymbol{X}), \boldsymbol{X} = \{X_1, X_2, X_3, X_4\}$, and assume that at some iteration the values of these variables are such that X_1 and X_2 naturally belong together and should not be split. Due to this dependence, they are consequently joined in a block (X_1, X_2). We draw the values for (X_1, X_2) (jointly) from the conditional $\Pr(X_1, X_2|x_3, x_4)$, where (x_3, x_4) are values from a previous iteration. In the next iteration, the situation may be such that the variables (X_2, X_3, X_4) inherently belong together, and consequently should be joined, and sampled as a block from $\Pr(X_2, X_3, X_4|x_1)$, and so on. As long as each X_i is sampled "infinitely often", jointly or individually, samples from $\Pr(\boldsymbol{X})$ are obtained in the limit.

3. Learning models via MCMC

MCMC offers a feasible way of learning BN models from a Bayesian perspective. In Chapter 3, Section 3.2, model learning was treated from a model selection point of view. To a Bayesian, a single model is a rather poor summary statistic of the posterior model distribution. A Bayesian is interested in the entire distribution $\Pr(M|\boldsymbol{d})$. Also, if we are interested in some feature over models quantified by Δ, we can average the feature with respect to the model posterior:

$$\mathrm{E}[\Delta(M)|\boldsymbol{d}] = \sum_m \Delta(m) \cdot \Pr(m|\boldsymbol{d})$$

In this section we discuss two methods for learning models via MCMC using the marginal likelihood scoring criterion; the discussion is based on Riggelsen, 2005. Other MCMC approaches for learning BN models exist, see for instance Friedman and Koller, 2003. An MCMC approach for model learning which does not employ the marginal likelihood criterion is given in Green, 1998. An alternative to MCMC is described in Madigan and Raftery, 1994.

The posterior distribution $\Pr(M|\boldsymbol{d})$ is obtained via Bayes' law, hence has the form of eq. 4.7, thus is precisely an instance of what can be solved via MCMC. The denominator $\sum_m \Pr(\boldsymbol{d}|m) \Pr(m)$ in eq. 3.12 is due to the large number of models impossible to evaluate.

3.1 Sampling models

MCMC Metropolis-Hastings sampling of models is discussed in Madigan and York, 1995; Kocka and Castelo, 2001. Here the *proposal distribution*, guides the incremental changes of the models by proposing to somehow change the current model. This proposal is produced by drawing $M' \sim \Pr'(M|M^{(t)})$. With probability:

$$\rho(M^{(t)}, M') = \min\left\{1, \frac{\Pr'(M^{(t)}|M')\Pr(M'|\boldsymbol{d})}{\Pr'(M'|M^{(t)})\Pr(M^{(t)}|\boldsymbol{d})}\right\}$$

the proposal is accepted and $M^{(t+1)} = m'$, otherwise $M^{(t+1)} = m^{(t)}$. For $t \to \infty$ models from the invariant distribution are obtained.

The usual proposal distribution changes the current model in a single adjacency, by selecting two vertices at random, and either adds, reverses or removes the edge (arc) between the vertices. Since these proposals are uniform, the proposal fraction $\Pr'(M^{(t)}|M')/\Pr'(M'|M^{(t)})$ cancels out, and the sampler is driven by the marginal likelihood ratio. Furthermore, since models consecutive differ in a single edge, it follows from the decomposition of the marginal likelihood, that all terms cancel out except those terms for which the parent set changes.

3.2 Sampling edges

In the following we apply a single-site Gibbs MCMC for sampling models from the posterior model distribution. Instead of considering M as a single random variable, we suggest to split the model into separate edges, each of which we regard as a random variable. This is the natural decomposition of a DAG model for applying Gibbs sampling, and is equivalent to applying the single adjacency Metropolis-Hastings sampler discussed in the previous section. However, the Gibbs sampling technique explicitly gives rise to the question if single edges is the best decomposition from an efficiency point of view: how does it influence the mixing of the chain? This is investigated in the sections to come, beginning from Section 3.3.

Define for all $r = 1, \ldots, \frac{p\cdot(p-1)}{2}$ possible edges of DAG model M the random variables E_r with state space $\Omega_{E_r} = \{\leftarrow, \rightarrow, \not\leftrightarrow\}$, i.e., every edge of the graph can take on a direction, or can be absent. If the configuration of all edges forms a DAG, the posterior joint distribution, $\Pr(E_1, \ldots, E_{\frac{p\cdot(p-1)}{2}}|\boldsymbol{d})$ is well-defined.

Via Gibbs sampling, models from the posterior are obtained as joint realisations of the edge assignments. The process goes as follows: draw

edges at iteration t from the full conditional given the data:

$$\begin{aligned} E_1^{(t)} &\sim \Pr(E_1 | e_2^{(t-1)}, \ldots, e_{\frac{p\cdot(p-1)}{2}}^{(t-1)}, \boldsymbol{d}) \\ &\vdots \\ E_{\frac{p\cdot(p-1)}{2}}^{(t)} &\sim \Pr(E_{\frac{p\cdot(p-1)}{2}} | e_1^{(t)}, \ldots, e_{\frac{p\cdot(p-1)}{2}-1}^{(t)}, \boldsymbol{d}) \\ E_1^{(t+1)} &\sim \Pr(E_1 | e_2^{(t)}, \ldots, e_{\frac{p\cdot(p-1)}{2}}^{(t)}, \boldsymbol{d}) \\ &\vdots \end{aligned}$$

Each draw is subject to the constraint that all edges together must form an acyclic graph. There are always at least two possible edge assignments. Removal is always an option since it can't introduce a cycle. Also, it is always possible to add an arc in at least one direction between any two vertices that are not adjacent.

The fact that not all 3 edge assignments are allowed at every drawing stage, means that irreducibility can't be guaranteed by using the positivity argument, i.e., that $\Pr(E_l | \boldsymbol{E} \setminus \{E_l\}, \boldsymbol{d}) > 0$. However, the Markov chain defined here *is* irreducible, because at every draw the state $\neq$ is a possibility and this state never introduces a cycle. Hence, there is a non-zero probability of removing arcs that obstruct the addition of other edges in any direction in the graph (obstruct in the sense that the graph would become cyclic). Consequently all DAGs can be reached by "breaking down" the current DAG (a kind of backtracking) and rebuilding another one thereby reaching another DAG.

In order to draw edge E_l from the full conditional given the data we calculate:

$$\begin{aligned} \Pr(E_l | \boldsymbol{E} \setminus \{E_l\}, \boldsymbol{d}) &= \frac{\Pr(E_1, \ldots, E_l, \ldots, E_{\frac{p\cdot(p-1)}{2}} | \boldsymbol{d})}{\sum_{e_l} \Pr(E_1, \ldots, e_l, \ldots, E_{\frac{p\cdot(p-1)}{2}} | \boldsymbol{d})} \\ &= \frac{\Pr(\boldsymbol{d} | E_1, \ldots, E_l, \ldots, E_{\frac{p\cdot(p-1)}{2}})}{\sum_{e_l} \Pr(\boldsymbol{d} | E_1, \ldots, e_l, \ldots, E_{\frac{p\cdot(p-1)}{2}})} \end{aligned} \quad (4.11)$$

where a uniform model prior $\Pr(M)$ on the model space, and the denominator $\Pr(\boldsymbol{d}) = \sum_m \Pr(\boldsymbol{d}|m) \Pr(m)$ both cancel out. When drawing

an edge from the Gibbs sampler, say E_l, at most two terms are affected, namely the terms pertaining to the vertices of edge E_l; denote those vertices by X_i and X_j. It follows from the marginal likelihood decomposition given in eq. 3.18 that:

$$\Pr(\boldsymbol{d}|\boldsymbol{E}) = \prod_{r \neq \{i,j\}} \Pr(\boldsymbol{d}_r|\boldsymbol{d}_{pa(r)}, \boldsymbol{E}) \cdot \Pr(\boldsymbol{d}_i|\boldsymbol{d}_{pa(i)}, \boldsymbol{E}) \Pr(\boldsymbol{d}_j|\boldsymbol{d}_{pa(j)}, \boldsymbol{E})$$

such that all factors $\prod_{r \neq \{i,j\}} \Pr(\boldsymbol{d}_r|\boldsymbol{d}_{pa(r)}, \boldsymbol{E})$ cancel out in eq. 4.11 because the parent sets for the corresponding vertices are unchanged.

To approximate the expected value of model features, we use the empirical average:

$$\mathrm{E}[\Delta(M)|\boldsymbol{d}] \approx \frac{1}{n} \sum_{t=1}^{n} \Delta(m^{(t)}) \tag{4.12}$$

where n denotes the total number of samples from the Markov chain. Often this kind of averaging is done over features one can read directly off a model, e.g., Markov blanket features of vertices, but theoretically any statement that the model entails can be averaged.

3.3 Blocking edges

In Section 2.1.6, the impact of blocking on the performance of MCMC was treated. Therefore, we need to analyse if there are edges that inherently belong together, and should be considered jointly when sampling. Although the univariate edge sampler theoretically does what it is supposed to do, it is from a practical point of view advantageous to seek blocks that can improve the mixing of the chain and prevent it from getting trapped for long periods of time.

The state of a single edge E_l between X_i and X_j is determined by those terms in the marginal likelihood that correspond to X_i and X_j, that is, in eq. 3.18 the terms $\Pr(\boldsymbol{d}_i|\boldsymbol{d}_{pa(i)}, \boldsymbol{E})$ and $\Pr(\boldsymbol{d}_j|\boldsymbol{d}_{pa(j)}, \boldsymbol{E})$. These terms indirectly depend on $\boldsymbol{X}_{pa(i)}$ and $\boldsymbol{X}_{pa(j)}$ in terms of their realisations in the (fixed) data sample. The marginal likelihood does not decompose into independent terms of parent set variables, and every time the parent set changes for a particular vertex, the corresponding term in the marginal likelihood needs to be recomputed. This means that when drawing an edge state for E_l, the current parent sets of X_i and X_j constrain the distribution of E_l. It follows that E_l depends on all incoming edges (arcs) to X_i and X_j; this is a form of context dependence, because when those same edges are outgoing or absent, the dependence is not

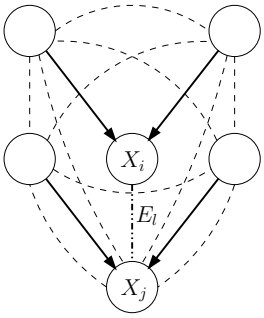

Figure 4.3. Dashed-dotted edge is the centre edge E_l. Solid edges (arcs) are the edges on which E_l currently depends; they "span" the block. All edges, including the dashed ones, potentially constrain each other (though in a varying degree) when assigned directions.

present. From the perspective of E_l, the dependence just described is a direct one, but the dependency goes even further, because the edges to X_i and X_j in turn depend on edges in the same way—this is an indirect dependence. When the parent sets $\boldsymbol{X}_{pa(i)}$ and $\boldsymbol{X}_{pa(j)}$ are large, E_l is influenced by many edges. This also means that in relatively dense areas of a DAG, the edges strongly depend on each other. Edge E_l can be considered the "centre" of a dependence region.

These considerations suggest that when we have to determine E_l, all edges in a region around E_l should be joined in a block, and be drawn "as one". First of all we notice that the dependence is a context dependence and therefore a block dynamically changes over time depending on the states of the edges in the vicinity of E_l. Also, the blocks overlap because the state of many edges is constrained by the same set of edges; this is especially so in dense regions where edges are close to each other. We suggest to apply a Gibbs sampler that *focuses per draw on one dependence region with one particular edge at the centre* instead of focusing on a single edge. By considering a region around E_l, we acknowledge that these edges may constrain the assignment of the state of the centre edge strongly, and to a lesser extent constrain each other.

The question that arises is then how large, or, alternatively, how far from the centre edge E_l, edges should be joined in a block. We suggest to join all edges on which E_l *directly* depends (along with E_l itself) into block l. However, this set of edges only determines the *current* situation but not necessarily "the right one" which is exactly what is object to learning. Therefore we also include in the block all edge variables *between*

the vertices spanned by the currently dependent edges, i.e., not only those pointing to X_i and X_j; see figure 4.3 for an illustration. All the edges in that block are "close" and either currently constrain each other (and most strongly constrain E_l), or potentially will constrain each other if assigned a direction.

A consequence of this dynamic way of blocking, is that areas that are currently dense, receive more attention and are sampled more frequently than less dense regions. In dense regions edges constrain each other, hence spending more time in these regions helps solving unfortunate edge assignments that negatively impact the edge configurations of dependent edges.

3.3.1 Blocks and Markov blankets

The suggested way of blocking coincides with the Markov blanket: the solid edges (arcs) in figure 4.3 connect X_i with the vertices in the Markov blanket of X_i. Hence, initially a block is defined by the edges in the Markov blanket of X_i.

The Markov blanket contains exactly those variables are the "most relevant" vertices for determining X_i. For that reason, it is reasonable to think that these variables potentially are related (adjacent in the graph) as well, since they represent concepts that are "close" from a domain perspective. Of course, this argument only is valid in the heuristic sense. The dashed edges in figure 4.3 can be thought of as representing those edges that potentially connect the Markov blanket vertices. In that respect including those edges in the block a priori is not a bad idea. We refer to the edges within a block as the *currently relevant* edges, and those not within the block as *currently irrelevant* edges.

Formally, these sets of edges are defined in the following way:

- The set of *relevant edges* of X_i is:

$$\boldsymbol{\mathcal{E}}'_i = \{E = (X_s, X_l) | X_s, X_l \in \boldsymbol{X}_{mb(i)} \cup \{X_i\}\}$$

- The set of *irrelevant edges* of X_i is:

$$\boldsymbol{\mathcal{E}}''_i = \{E = (X_i, X_l) | X_l \in \boldsymbol{X}\} \setminus \boldsymbol{\mathcal{E}}'_i$$

- A *block* of edges for X_i is: $\boldsymbol{\mathcal{E}}_i = \boldsymbol{\mathcal{E}}'_i \cup \boldsymbol{\mathcal{E}}''_i$

In figure 4.4 the 3 sets are illustrated.

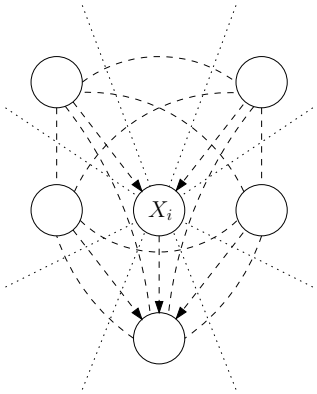

Figure 4.4. A block, $\boldsymbol{\mathcal{E}}_i$. Dashed edges belong to the relevant edges, $\boldsymbol{\mathcal{E}}_i'$. Dotted edges belong to the irrelevant edges, $\boldsymbol{\mathcal{E}}_i''$.

The general approach is now as follows: either shrink or grow the Markov blanket of a vertex by adding or removing vertices via currently irrelevant edges, *or* change the internal relationships between the vertices of the Markov blanket via the currently relevant edges. Observe that $\boldsymbol{\mathcal{E}}_i'$ actually captures the notion of a sub-graph, and that the proposed approach boils down to learning sub-graphs that, when combined, form the overall DAG model.

3.3.2 Sampling blocks

Block Gibbs sampling in terms of $\boldsymbol{\mathcal{E}}_i$, follows the sampling scheme:

$$\begin{aligned}
\boldsymbol{\mathcal{E}}_1^{(t)} &\sim \Pr(\boldsymbol{\mathcal{E}}_1 | \bar{\boldsymbol{\varepsilon}}_1, \boldsymbol{d}) \\
&\vdots \\
\boldsymbol{\mathcal{E}}_p^{(t)} &\sim \Pr(\boldsymbol{\mathcal{E}}_p | \bar{\boldsymbol{\varepsilon}}_p, \boldsymbol{d}) \\
\boldsymbol{\mathcal{E}}_1^{(t+1)} &\sim \Pr(\boldsymbol{\mathcal{E}}_1 | \bar{\boldsymbol{\varepsilon}}_1, \boldsymbol{d}) \\
&\vdots
\end{aligned}$$

Here $\bar{\boldsymbol{\varepsilon}}_j$ is the *current* configuration of the edges in the *complement* of the set $\boldsymbol{\mathcal{E}}_j$. As previously remarked in Section 2.1.4, the visitation scheme is not crucial, and random selection of blocks may be beneficial for computational reasons.

Sampling a block of edges may seem difficult at first sight. In contrast to single edges, entire blocks have many different configurations. The probability of each possible configuration of $\boldsymbol{\mathcal{E}}_i$ needs to be determined before we can draw one. Additionally, the relevant and irrelevant edges play different roles. Adding a vertex not currently belonging to the Markov blanket via the irrelevant edges should have a lower priority than changing the internal relationships between those vertices that already belong to the Markov blanket via the relevant edges.

Metropolis-Hastings sampling provides a way to deal with these issues. We use a "Metropolis-within-Gibbs" MCMC sampler, where Gibbs sampling and Metropolis-Hastings sampling are combined in order to sample models from the invariant model posterior. Each draw from the Gibbs sampler is performed by a Metropolis-Hastings sampler. The proposal distribution is defined as a mixture distribution, where one component, $f(\cdot)$, deals with the edges in the relevant edge set, and the other component, $g(\cdot)$, deals with the edges in the irrelevant edge set, that is:

$$\Pr'(\boldsymbol{\mathcal{E}}^{(t+1)}|\boldsymbol{\mathcal{E}}^{(t)}) = w \cdot f(\boldsymbol{\mathcal{E}}^{(t+1)}|\boldsymbol{\mathcal{E}}'^{(t)}) + (1-w) \cdot g(\boldsymbol{\mathcal{E}}^{(t+1)}|\boldsymbol{\mathcal{E}}''^{(t)}) \quad (4.13)$$

where $0 < w < 1$ determines the mixture weights.

When $f(\cdot)$ is applied, values for the edges in the relevant set are drawn, i.e., MCMC is run in order to obtain the posterior distribution of those edges. The cardinality of the set of relevant edges is kept fixed once the Metropolis-Hastings sampler is applied—we merely assign (new) values to the edge variables. Hence, the set of relevant edges is determined before entering the Metropolis-Hastings sampler, and does not change until control is given back to the overall Gibbs sampler. For $g(\cdot)$ the same holds, but here assignments are considered to the variables in the irrelevant set.

The distribution $f(\cdot)$ produces uniform proposals such that each edge $E_r \in \boldsymbol{\mathcal{E}}'$ has probability $1/|\boldsymbol{\mathcal{E}}'|$ of being drawn. Depending on the current value e_r, a state change is proposed to one of the (at most) two alternatives with probability 0.5. For example, if $E_r^{(t)} = \neq$ then either $E_r^{(t+1)} = \rightarrow$ or $E_r^{(t+1)} = \leftarrow$ is proposed. For the distribution $g(\cdot)$ the same holds, but here we have $E_r \in \boldsymbol{\mathcal{E}}''$ with probability $1/|\boldsymbol{\mathcal{E}}''|$. Notice that edges not in either of these two sets remain unchanged.

The weight w in the mixture varies how much "attention to pay" to the configuration of the edges in the Markov blanket of the current vertex. We want to try out several different configurations of the edges in the relevant edge set before deciding to grow or shrink it via the

irrelevant edge set. This implies that $f(\cdot)$ should be applied more often than $g(\cdot)$.

Uniform proposals are not the only option, but it makes the Metropolis-Hastings acceptance ratio easy to compute. For Metropolis-Hastings, the acceptance probability depends on the following fraction, with the proposal distribution defined in eq. 4.13:

$$\frac{\Pr'(\boldsymbol{\mathcal{E}}^{(t)}|\boldsymbol{\mathcal{E}}^{(t+1)})}{\Pr'(\boldsymbol{\mathcal{E}}^{(t+1)}|\boldsymbol{\mathcal{E}}^{(t)})}$$

For both the mixture in the numerator and in the denominator, the weights are the same, the conditional distributions select edges with equal probability and there is always the same number of alternative edge assignments, i.e., the distributions are uniform, hence, the ratio cancels out.

If non-uniform proposals are used, then the proposal ratio does not cancel out, and it has to be computed explicitly. It is questionable if one is able to define such non-uniform proposals unless prior knowledge is available. Moreover, prior knowledge is rarely expressed in terms of relevant and irrelevant edge sets, so specifying a non-uniform proposal distribution is in no way trivial.

There seems to be something counterintuitive as to why the combined MCMC sampler approach works: why sample single edges via Metropolis-Hastings when the performance gain lies in the fact that all edges in a block should be considered "one entity"? One could argue that the problem with the edge dependences has just been pushed down one level to another MCMC sub-process, and that this process suffers from exactly the same thing as the original single edge Gibbs MCMC. The reason why the proposed "Metropolis-within-Gibbs" approach works is that edges that are part of several Markov blankets are sampled relatively often. This entails that edges in dense regions of the DAG are sampled more often. More time is spent in regions where edges constrain each other strongly, meaning that potential obstructions are "solved" simply because improbable yet not impossible alternatives edge assignments are eventually accepted because they have been proposed many times. In doing so we take into account that sampling models is more difficult in some parts of the model space.

3.3.3 Validity of the sampler

In using such a "Metropolis-within-Gibbs" sampler, the question arises if the convergence towards the invariant model distribution is still guar-

anteed. Combining MCMC samplers boils down to combining several transition probabilities via a mixture or product of transitions. It is well-known (Tierney, 1994) that such combinations leave the overall chain ergodic with the desired invariant distribution, provided the sub-MCMC samplers created via Metropolis-Hastings sampling in the separate Gibbs sampling steps are ergodic with the invariant distributions $\Pr(\boldsymbol{\mathcal{E}}_i|\bar{\boldsymbol{\mathcal{E}}}_i, \boldsymbol{d})$.

THEOREM 4.1 *The* MB-MCMC *model sampler produces an ergodic Markov chain with the invariant distribution* $\Pr(M|\boldsymbol{d})$.

It is not difficult to see that this is the case, because it follows from the fact that Metropolis-Hastings MCMC produces an ergodic Markov chain: The proposal distribution of the Metropolis-Hastings sampler will with a non-zero probability propose a state change to any edge in $\boldsymbol{\mathcal{E}}_i$, which guarantees irreducibility. With a non-zero probability it will remain in the current state for any edge implying aperiodicity. We may thus conclude that the Metropolis-Hastings sampler, for all $i = 1, \ldots, p$, in the limit returns realisations from the invariant distribution $\Pr(\boldsymbol{\mathcal{E}}_i|\bar{\boldsymbol{\mathcal{E}}}_i, \boldsymbol{d})$, i.e., realisations for the edges in $\boldsymbol{\mathcal{E}}_i$ given all other edges. We note that formally it is not even an requirement that Markov chains created by the Metropolis-Hastings samplers converges for every i before going on to the next Gibbs sampling step.

Since blocks dynamically change over time, we need to check if all edges are sampled. The Gibbs sampler makes a call to a Metropolis-Hastings sampler for every vertex, and it trivially follows that every edge eventually will be part of an irrelevant edge set. We can't guarantee that the edges will be part of a relevant edge set however, but as long as all edges *are* considered (and sampled via Metropolis-Hastings MCMC), we have that the realisations drawn come from $\Pr(E_1, \ldots, E_{\frac{p\cdot(p-1)}{2}}|\boldsymbol{d})$ as $t \to \infty$.

3.3.4 The MB-MCMC model sampler

Algorithm 1 contains the pseudocode of the Markov blanket MCMC (MB-MCMC) sampler. Line 3 determines the block to pay attention to; here a systematic sweep is shown. Line 4 calls the algorithm for reversing covered arcs; we refer to Kocka and Castelo, 2001 for the implementation of this step. Lines 5–6 determines the edges to consider, and in lines 8–12 the edges are drawn from the sets of relevant edges. The proposals are accepted or rejected in line 13–16. In line 17 the configuration of all edges is recorded, i.e., here the actual models from the posterior are

saved. One may decide to sub-sample the Markov chain of models by only recording the draws once in a while.

Algorithm 1: MB-MCMC

Input : k, MH-steps; w, prop. expand/change block conf.
Output: Edges from $\Pr(\boldsymbol{E}|\boldsymbol{d})$ (requires burn-in).

$M \leftarrow G = (\boldsymbol{X} = \{X_1, \ldots, X_p\}, \boldsymbol{E} = \{E_1 =\not\sim, \ldots, E_{\frac{p\cdot(p-1)}{2}} =\not\sim\})$
for $r \leftarrow 0$ **to** ∞ **do**
  $i \leftarrow (r \bmod p) + 1$
  `rcar(`*10*`)`
  `/* Define relevant and irrelevant sets/blocks */`
  $\boldsymbol{\mathcal{E}}'_i \leftarrow \{E = (X_s, X_l) | X_s, X_l \in \boldsymbol{X}_{mb(i)} \cup \{X_i\}\}$
  $\boldsymbol{\mathcal{E}}''_i \leftarrow \{E = (X_i, X_l) | X_l \in \boldsymbol{X}\} \setminus \boldsymbol{\mathcal{E}}'_i$
  for $t \leftarrow 0$ **to** k **do**
    draw $U \sim \mathcal{U}[0,1]$ `/* Uniform draw */`
    `/* Draw edge values for the block */`
    if $u < w$ **and** $\boldsymbol{\mathcal{E}}'_i \neq \varnothing$ **then**
      draw $\boldsymbol{\mathcal{E}}_i^{(t+1)} \sim f(\boldsymbol{\mathcal{E}}_i | \boldsymbol{\varepsilon}_i'^{(t)})$
    else
      draw $\boldsymbol{\mathcal{E}}_i^{(t+1)} \sim g(\boldsymbol{\mathcal{E}}_i | \boldsymbol{\varepsilon}_i''^{(t)})$
    $\rho \leftarrow \min\{1, \Pr(\boldsymbol{d}|\boldsymbol{\varepsilon}_i^{(t+1)}) / \Pr(\boldsymbol{d}|\boldsymbol{\varepsilon}_i^{(t)})\}$
    draw $U \sim \mathcal{U}[0,1]$
    if $u \geq \rho$ **then**
      $\boldsymbol{\mathcal{E}}_i^{(t+1)} \leftarrow \boldsymbol{\varepsilon}_i^{(t)}$
  `record(`$e_1, \ldots, e_{\frac{p\cdot(p-1)}{2}}$`)`

The algorithm takes two arguments: k determines the number of times the Metropolis-Hastings sampler is run, and w determines the probability of changing the internal configuration of a component vs. adding or removing new vertices. Parameter k need not be large for the overall invariant model distribution to be reached, i.e., the Metropolis-Hastings sampler need not converge at every call. In fact we have found it to be beneficial for the convergence rate to assign k a small value; too large a value may lead to slow mixing and convergence. In our experiments we have set $k = 5$, and $w = 0.95$. Letting the Metropolis-Hastings sampler converge before the overall Gibbs sampler reaches its invariant distribution, means that the invariant distributions of the Metropolis-Hastings

samplers are reached and they now coincide with the marginal (at that time still) sub-optimal distribution of the Gibbs sampler. There is no reason to exhaustively explore the state space of the marginals of the Gibbs sampler, unless we are close to its invariant distribution. Doing so has a negative impact on the acceleration of the chain.

When every vertex is assigned a cache that keeps the sufficient statistics indexed by the parent set, we may drastically improve the speed of MCMC by querying the cache before querying the data. We have implemented the Markov blanket sampler in C++ using STL, and for the experiments in the next section we were able to reach what we believe are the invariant distributions in less than 10 minutes on a 2 GHz machine.

3.3.5 Evaluation

We considered two BNs for the experiments: The ALARM network with 37 vertices and 46 arcs (Beinlich et al., 1989) about intensive care patient monitoring, and the Insurance network with 27 vertices and 52 arcs (Binder et al., 1997) classifying car insurance applications. We used the BDeu metric for the counts $\boldsymbol{\alpha}$ with an ESS of 1. All experiments were run for 1,000,000 iterations. As convergence diagnostic we monitored the number of edges as suggested in for instance Giudici and Green, 1999. We compared the Markov blanket MCMC with $e\text{MC}^3$, a single edge MCMC sampler that also employs the RCAR algorithm (Kocka and Castelo, 2001; Castelo, 2002).

In figure 4.5 the results of the ALARM network are illustrated for 1000 and 5000 samples. With 1000 samples, we see that two independent runs of the MB-MCMC both converge towards models with about 50–53 edges. There is no significant difference in the convergence behaviour. For $e\text{MC}^3$ two runs produce different behaviour and result in models with 68–77 edges. For 5000 records similar observations hold, but overall the number of edges is lower: 45–51 for MB-MCMC and 57–70 for $e\text{MC}^3$. We notice that $e\text{MC}^3$ seems sensitive to the starting point of the chain. To show this more clearly, we ran both samplers starting from the empty graph, and from the actual ALARM graph for 7000 samples. The results are illustrated in figure 4.6. For the 7000 records we would expect that the number of edges on average should converge to 46, i.e. there is enough data to support the data generating model. For MB-MCMC, both chains converge towards models with 44–50 edges. The most frequently sampled model is similar to the ALARM network ± 2 arcs. For $e\text{MC}^3$ there is a big difference.

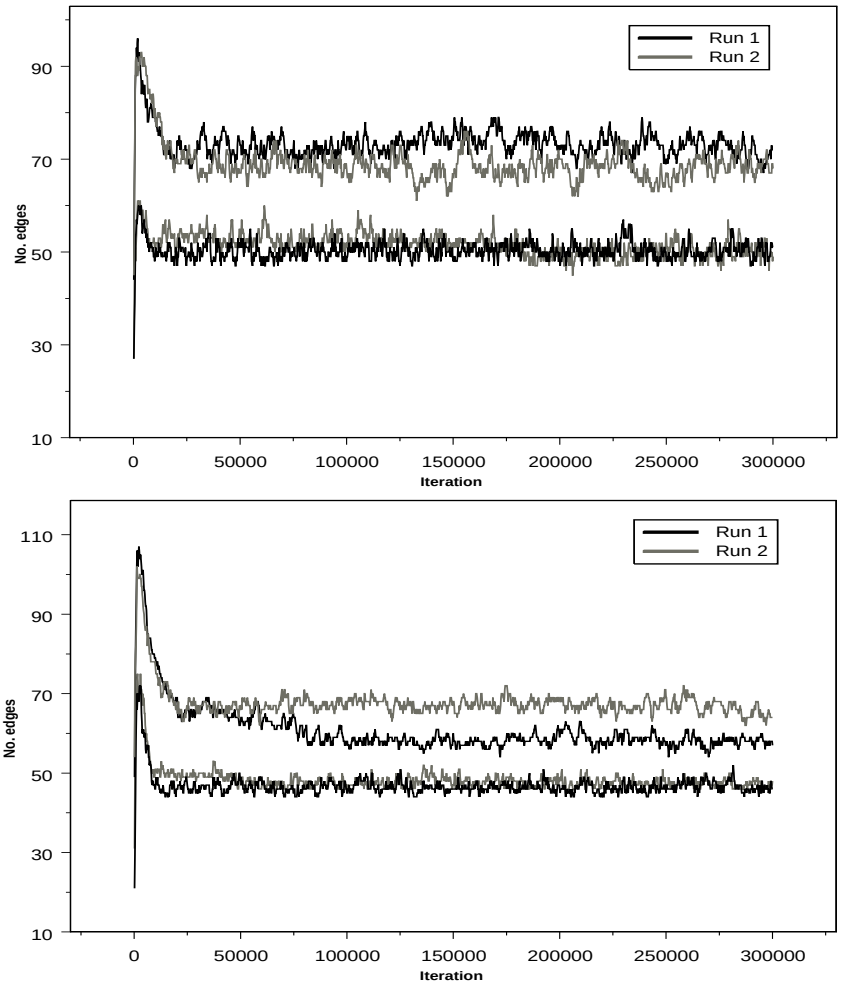

Figure 4.5. ALARM network. Convergence behaviour given 1000 (*top*) and 5000 (*bottom*) records for two independent runs. The lower lines are from the Markov blanket MCMC, and the upper lines from $e\text{MC}^3$.

The chain started from the actual network stays at around 50–55 edges, but the chain started from the empty graph gets stuck at 63–70. The most frequently sampled model is in both situations less similar to the actual ALARM network than in the MB-MCMC case (excess of ± 10 and ± 25 arcs).

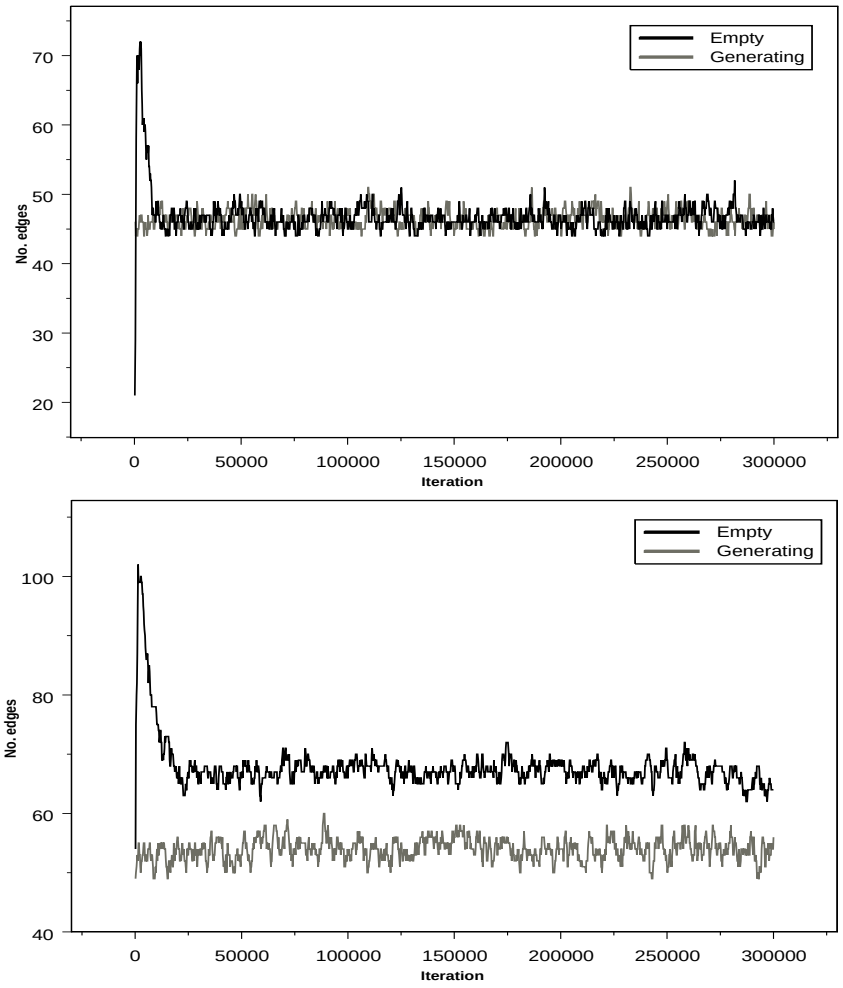

Figure 4.6. ALARM network. Convergence behaviour of the Markov blanket MCMC (*top*) and $e\text{MC}^3$ (*bottom*) given 7000 records starting from the empty and the data generating model.

Next we consider results of the Insurance network in figure 4.7 for 500 samples. We would like to note that the association between several parent-child variables in the Insurance network is rather weak and that even for large data sets these associations will be deemed absent by the marginal likelihood score. For 500 records the MB-MCMC converges to an invariant distribution where models are sampled with 36–40 edges.

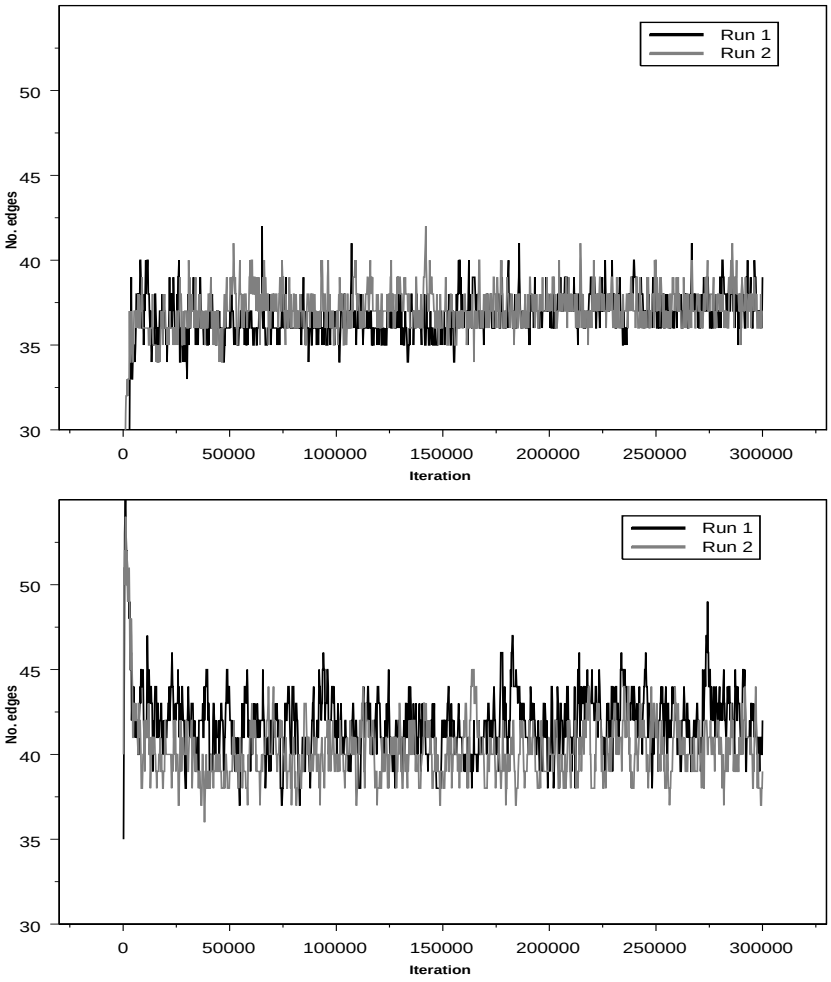

Figure 4.7. Insurance network. Convergence behaviour given 500 records for two independent runs of the Markov blanket MCMC (*top*) and eMC^3 (*bottom*).

The two runs meet at around 150,000 iterations. For eMC^3 however, the two chains don't quite agree in the number of edges: somewhere between 37–46. We also ran both samplers beginning from the empty and the actual Insurance graph for which the results are illustrated in figure 4.8. For MB-MCMC both starting points produce models with 45–47 edges. Also here we see that eMC^3 is sensitive to the initial model. Starting

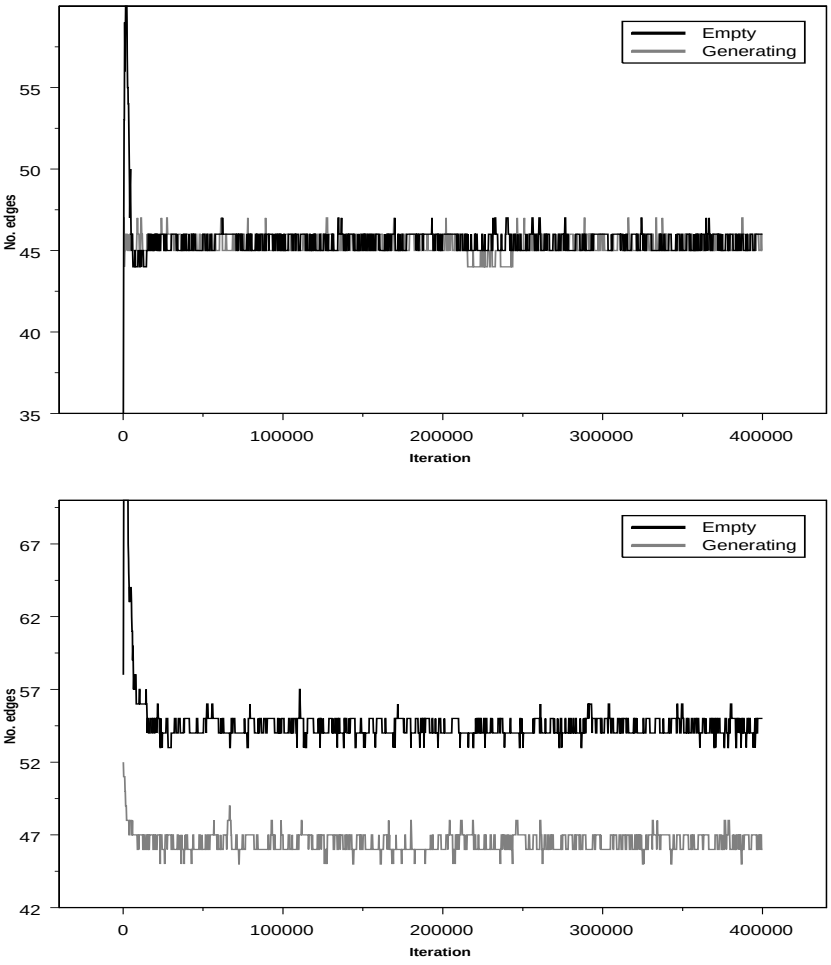

Figure 4.8. Insurance network. Convergence behaviour of the Markov blanket MCMC (*top*) and $e\text{MC}^3$ (*bottom*) given 10,000 records starting from the empty and the data generating model.

from the data generating model, the sampler converges to an invariant distribution where models with 45–47 edges are sampled. Starting from the empty graph, models with 54–56 edges are sampled. We see that even with 10,000 records, there is not enough information in the data sample to support the 52 arcs in the data generating Insurance network.

Observe that for large amounts of data, the fluctuation in the number of edges is larger. This is especially noticeable when comparing the plots in figures 4.7 and 4.8; the variability of the plots for 500 records is larger than for 10,000 records. This is to be expected because there is no pronounced "best" model with merely 500 records. For large amounts of data, the models do not differ a lot, and model selection may in that case be an efficient alternative to MCMC. After all, MB-MCMC is from a computational point of view generally more demanding than most algorithms developed for model selection.

3.3.6 Conclusion

MB-MCMC is a Bayesian approach to learning Bayesian network models. Being Bayesian is beneficial when model learning is based on a relatively small amount of data; in practice this may often be the case. When data is scarce, there may be several models that are structurally quite different from each other, yet are almost equally likely given the data. In other words, the data supports several models that differ widely, yet from a scoring perspective are very close. Model selection methods will only return "the best" model, but give no clue as to how and in what respect models differ that score almost equally well.

For large data sets—where "large" of course is related to the number of variables of our domain—the best model is much more likely than any other model, and model selection may be adequate.

By employing MB-MCMC there is no added computational burden compared to existing MCMC approaches. The improvement in mixing and convergence is only due to a wiser decomposition of the joint distribution. From a local perspective, the edges in the Markov blanket of the vertices form a natural dependence relationship in the sense that they constrain each other. This local dependence also from a more global perspective makes sense: dense areas of the DAG is "tougher" to alter than less dense regions.

Chapter 5

LEARNING FROM INCOMPLETE DATA

Real-life data is rarely complete. Unfortunately most learning algorithms assume that the data is complete, or handle missing values in an ad-hoc way. In this chapter we introduce the concept of incomplete data and how to learn BNs from incomplete data. We illustrate why this is a non-trivial problem, and why existing complete data methods will not work "out of the box" anymore. Incomplete data often renders an exact analysis intractable making approximation or iteration the only feasible approach to such problems.

We present various principled approaches that require iteration and apply the theory treated in the previous chapters to learning BNs from incomplete data. We focus on the Bayesian approach to learning BNs and several MCMC samplers are presented for approximating posterior distributions. In particular, we develop an algorithm called eMC^4 that combines importance sampling, Gibbs sampling and model MCMC sampling for drawing from the posterior model distribution given incomplete data. Less principled non-iterative approaches are discussed as well; they are fast, but only approximate results are obtained. Towards the end of this chapter a non-iterative algorithm is developed, called MBP. By relaxing assumptions about interactions between variables and by exploiting the "blocking influence from outside" effects of Markov blankets, the algorithm is very fast, but still accurate in comparison with iterative approaches.

1. The concept of incomplete data

Although "missing data" and "incomplete data" usually are synonymous, in the current context the latter is preferred, because it suggests that a completion of the data exists—somehow the complete data has been censored or otherwise restricted from being fully observed. In fact the notion of there existing a potential completion of the incomplete data is central to the analysis of such data.

To get an intuitive idea of the concept of incomplete data, consider the following example from credit scoring (see for instance Hand and Henley, 1997) about applicants applying for a loan at a bank. The applicants are either rejected or accepted depending on characteristics of the applicant such as age, income and marital status. The vector $\mathbf{X} = (X_1, \ldots, X_k)$ refers to those completely observed application variables. Repayment behaviour of the accepted applicants is observed by the creditor, usually leading after some time to a classification as either a good or bad (defaulted) loan; let the binary variable Y be this classification label. Repayment behaviour of rejects is for obvious reasons not observed; complete data is available only for accepted applicants. The response indicator R_Y, indicates if the applicant was accepted or rejected.

We can write the distribution of the outcome of the loan as:

$$\Pr(Y|\mathbf{X}) = \sum_{i=0}^{1} \Pr(Y|\mathbf{X}, R_Y = i) \Pr(R_Y = i|\mathbf{X})$$

Note that the sampling process identifies the acceptance/rejection probability $\Pr(R_Y|\mathbf{X})$ and the outcome conditional on acceptance $\Pr(Y|\mathbf{X}, R_Y = 1)$, but provides no information about the outcome conditional on rejection $\Pr(Y|\mathbf{X}, R_Y = 0)$. Hence we have a problem in estimating $\Pr(Y|\boldsymbol{X})$ from *observed* data.

Now suppose the acceptance policy is such that the probability of acceptance depends exclusively on $\mathbf{X}$; for example, values of $\mathbf{X}$ corresponding to a high perceived default risk have a low acceptance probability (and vice versa), but otherwise acceptance is completely random. Then we have:

$$\Pr(Y|\mathbf{X}, R_Y = 1) = \Pr(Y|\mathbf{X}, R_Y = 0) = \Pr(Y|\mathbf{X})$$

Now $\Pr(Y|\mathbf{X})$ coincides with the observable distribution $\Pr(Y|\mathbf{X}, R_Y = 1)$, which we can therefore use to estimate $\Pr(Y|\mathbf{X})$. In other words,

valid statistical analysis can be performed by considering the observed part of the data only.

Suppose instead that acceptance is based on additional characteristics $\boldsymbol{Z}$ other than $\boldsymbol{X}$, that is $\boldsymbol{Z} \not\subseteq \boldsymbol{X}$, for example the "general impression" that the loan officer has of the applicant. If $\Pr(Y|\boldsymbol{X}, \boldsymbol{Z}) \neq \Pr(Y|\boldsymbol{Z})$, that is Y indeed depends on $\boldsymbol{Z}$ even when we condition on $\boldsymbol{X}$, this leads to:

$$\Pr(Y|\boldsymbol{X}, R_Y = 1) \neq \Pr(Y|\mathbf{X}, R_Y = 0) \tag{5.1}$$

i.e., at any particular value of $\mathbf{X}$, the distribution of the observed Y differs from the distribution of the missing Y. Note that if the loan officer does his job well, we expect that $\Pr(Y = 1|\boldsymbol{X}, R_Y = 1) > \Pr(Y = 1|\mathbf{X}, R_Y = 0)$ (where $Y = 1$ denotes a good loan). The inequality in eq. 5.1 means that we cannot perform valid statistical analysis by only considering the observed part of the data.

1.1 Missing data mechanisms

As we saw in the previous example, the variable R_Y is used to indicate if Y is observed or not. The distribution of R_Y models the process giving rise to incompletion. This mechanism may depend on observed and possibly on unobserved entries as well.

In general we consider the vector $\boldsymbol{R}$ that has the same dimensions as $\boldsymbol{D}$, where R_i^j is the binary indicator variable of X_i^j, i.e., for record j, variable X_i. When X_i^j is missing, then $R_i^j = 0$, otherwise $R_i^j = 1$. Additionally we partition $\boldsymbol{D} = (\boldsymbol{O}, \boldsymbol{U})$, where $\boldsymbol{O}$ is the observed part and $\boldsymbol{U}$ the unobserved (missing) part.

Rubin, 1976, defines three basic missing data mechanisms: *Missing At Random* (MAR), *Missing Completely At Random* (MCAR) and *Missing Not At Random* (MNAR). They are defined as follows:

MAR $\Pr(\boldsymbol{R}|\boldsymbol{O}, \boldsymbol{U}) = \Pr(\boldsymbol{R}|\boldsymbol{O})$, i.e., given $\boldsymbol{O}$ the probability that an observation is missing does not depend on $\boldsymbol{U}$.

MCAR $\Pr(\boldsymbol{R}|\boldsymbol{O}, \boldsymbol{U}) = \Pr(\boldsymbol{R})$, i.e., the probability that an observation is missing depends neither on $\boldsymbol{O}$ nor $\boldsymbol{U}$. It follows that this is a special case of MAR.

MNAR The probability that an observation is missing depends on $\boldsymbol{U}$ (and possibly on $\boldsymbol{O}$). MAR and MNAR form disjoint classes—a missing data mechanism is either MAR or MNAR.

In general we are interested in the so-called *predictive distribution* $\Pr(\boldsymbol{U}|\boldsymbol{O})$, which is based solely on the observed part of the data:

$$\Pr(\boldsymbol{U}|\boldsymbol{O}) = \sum_{\boldsymbol{r}} \Pr(\boldsymbol{U}|\boldsymbol{O},\boldsymbol{r}) \Pr(\boldsymbol{r}|\boldsymbol{O}) \tag{5.2}$$

The MAR assumption exactly formalises the idea of being able to predict ("restore") the missing data, when response is independent of missing values. From the MAR assumption it follows by swapping $\boldsymbol{U}$ and $\boldsymbol{R}$ in $\Pr(\boldsymbol{R}|\boldsymbol{O},\boldsymbol{U})$:

$$\Pr(\boldsymbol{U}|\boldsymbol{O},\boldsymbol{R}) = \Pr(\boldsymbol{U}|\boldsymbol{O})$$

That is, for predicting $\boldsymbol{U}$ we need not consider the response indicator variable for unbiased statistical analysis. We may now write eq. 5.2 as:

$$\sum_{\boldsymbol{r}} \Pr(\boldsymbol{U}|\boldsymbol{O},\boldsymbol{r}) \Pr(\boldsymbol{r}|\boldsymbol{O}) = \sum_{\boldsymbol{r}} \Pr(\boldsymbol{U}|\boldsymbol{O}) \Pr(\boldsymbol{r}|\boldsymbol{O})$$

and we say that the missing data mechanism is *ignorable*. Strictly speaking ignorability also requires *distinctness of parameters* (Little and Rubin, 1987). In Bayesian terms, this means that the parameter of the distribution of the complete data $(\boldsymbol{O},\boldsymbol{U})$ should be independent of the parameter of the distribution governing the missing data mechanism. In practice this usually holds, but if violated, it "only" comes at the price of efficiency, but the predictions remain unbiased.

If data is missing according to an MNAR mechanism, then the MAR independence statement does not hold, and we need to predict according to $\Pr(\boldsymbol{U}|\boldsymbol{O},\boldsymbol{R})$. In that case the missing data mechanism has to be modeled explicitly, for instance by someone with domain knowledge. In the latter part of the example given in Section 1, the acceptance (rejection), depends on an unobserved set of characteristics, imposed by the "domain expert"; this clearly violates the MAR assumption, because $\Pr(R_Y|\boldsymbol{O}=\boldsymbol{X},\boldsymbol{U}=\boldsymbol{Z}) \neq \Pr(R_Y|\boldsymbol{O}=\boldsymbol{X})$. If $\boldsymbol{Z}$ would have been observed in conjunction with $\boldsymbol{X}$, then the missing data mechanism would have been MAR, because the probability of acceptance is no longer dependent on covariates that are unobserved. This suggests that when a missing data mechanism is MNAR it in fact may be become MAR if additional data is obtained; in this case if we could get hold of $\boldsymbol{Z}$. Also, MAR vs. MNAR is really a matter of degree, since we may be able to get hold of a subset of $\boldsymbol{Z}$ thus coming "closer to MAR".

Unfortunately, it is not possible by analysing the incomplete data to determine if data is missing according to MAR or MNAR, unless we

have knowledge about the missing data mechanism. However, as we mentioned above, MAR and MNAR are the two extremes, i.e., between $\Pr(\boldsymbol{U}|\boldsymbol{O},\boldsymbol{R})$ and $\Pr(\boldsymbol{U}|\boldsymbol{O})$ there are several levels of dependence on $\boldsymbol{R}$. In fact $\Pr(\boldsymbol{U}|\boldsymbol{O},\boldsymbol{R}) \neq \Pr(\boldsymbol{U}|\boldsymbol{R})$ covers the entire interval between "almost equal" or "entirely different", e.g., with regard to context independence. Effectively this means that even though the MAR assumption may be violated to some extent, analysis may still yield approximately valid results. Although we cannot test for ignorability in general, an option is to test how *sensitive* inference results are to different missing data mechanisms.

In general it is impossible to develop fully automated learning procedures, when data is missing non-ignorable. In that case prior knowledge has to be specified about the nature of the missing data mechanism. In this thesis we develop methods that assume ignorability, in which case the missing data mechanism need not be specified.

Under the ignorability assumption, a valid (unbiased) prediction of $\boldsymbol{U}$ can be obtained based on $\boldsymbol{O}$. Thus, all information about the missing data, necessary for performing valid statistical analysis, is already contained in the observed data $\boldsymbol{O}$ prior to predicting $\boldsymbol{U}$; the predictions made for $\boldsymbol{U}$ do not produce information not already implicitly contained in $\boldsymbol{O}$. Combining $\boldsymbol{O}$ and the predictions of $\boldsymbol{U}$ merely makes the sample look complete, yet the actual sample size is really that of $\boldsymbol{O}$. In every respect it suffices to analyse $\boldsymbol{O}$ only, and not predict $\boldsymbol{U}$ at all.

Although all required information is implicitly contained in $\boldsymbol{O}$, the observed data may be structured in a way that complicates the extraction of this information. In an MCAR setting the structural difficulties can be circumvented at the price of loss of efficiency. MCAR implies that records with missing values can be skipped; since MCAR implies that the response indicator is independent of the data values, skipping records with missing values translates into sub-sampling the incomplete data. What remains is a complete sample, and any complete data statistical technique can be applied. In Section 4 we return to this issue. When data is not MCAR, then this process of skipping records obviously introduces bias, and another approach is required.

2. Learning from incomplete data

Principled statistical analysis of MAR incomplete data needs to consider all of $\boldsymbol{O}$, not just subsets as in an MCAR analysis.

The challenges and difficulties with incomplete data pervades all areas of statistical data analysis. An exact analysis is usually out of the questions because it leads to analytical intractability and high computational complexity. In the remainder of the thesis we restrict attention to learning BNs from incomplete data; see also Chickering and Heckerman, 1997 for interesting discussions on large sample approximations when learning BNs.

In the following, we are given a data sample with incomplete records. Each record contains instantiations of the variables such that for record j we have that $\boldsymbol{d}^j = (x_1^j, \ldots, x_p^j) = (\boldsymbol{o}^j, \boldsymbol{u}^j) = (\boldsymbol{o}^j, (u_1^j, \ldots, u_{r(j)}^j))$, where $\boldsymbol{o}^j$ is the observed part of the record, and $\boldsymbol{u}^j$ is the $r(j)$-dimensional unobserved part.

2.1 Likelihood decomposition

The likelihood function plays a crucial role in learning BNs, no matter if the classical (penalised) likelihood approach is used or the Bayesian approach.

The learning methods introduced in Chapter 3 rely on the fact that the functional form of the likelihood is a product of terms. For incomplete data this is unfortunately no longer true.

Via the likelihood decomposition given in eq. 3.2, it follows that the likelihood of observed data is given by:

$$\Pr(\boldsymbol{o}|m, \boldsymbol{\theta}) = \prod_{j=1}^{c} \sum_{\boldsymbol{u}^j} \prod_{i=1}^{p} \theta_{x_i^j | \boldsymbol{x}_{pa(i)}^j}$$

That is, the unobserved variable are summed out for each record: the sum is taken over all possible completions of the unobserved part. The likelihood is no longer a simple product, but includes summations as well. For a given record there will for observed variables be as many terms as there are completions of the ancestral variables in m. In general this means that the likelihood of incomplete data has a number of summation terms roughly exponential in the number of missing items in the sample.

As an example, suppose we are given the following model m with 3 binary variables, $X_1 \rightarrow X_2 \rightarrow X_3$, and assume that we are given the following incomplete data sample ('?' denotes a missing observation):

	X_1	X_2	X_3
$\boldsymbol{d}^1$	0	1	1
$\boldsymbol{d}^2$	1	?	1

After seeing $\boldsymbol{d}^1$, the likelihood $\Pr(\boldsymbol{d}^1|\boldsymbol{\theta}, m)$ is still a simple product (where $1 - \theta_{X_i=1|pa(i)} = \theta_{X_i=0|pa(i)}$):

$$(1 - \theta_{X_1=1})^1 \theta^1_{X_2=1|X_1=0} \theta^1_{X_3=1|X_2=1}$$

but once $\boldsymbol{d}^2$ is observed, we need to sum out X_2, and the likelihood $\Pr(\boldsymbol{d}^1, \boldsymbol{o}^2|\boldsymbol{\theta}, m)$ becomes:

$$(1 - \theta_{X_1=1})^1 \theta^1_{X_1=1} \theta^1_{X_2=1|X_1=0} \theta^1_{X_2=1|X_1=1} \theta^2_{X_3=1|X_2=1} +$$
$$(1 - \theta_{X_1=1})^1 \theta^1_{X_1=1} \theta^1_{X_2=1|X_1=0} (1 - \theta_{X_2=1|X_1=1})^1 \theta^1_{X_3=1|X_2=1} \theta^1_{X_3=1|X_2=0}$$

a summation over likelihoods of complete data, one per completion.

In rare cases the likelihood of incomplete data remains a product, namely if all descendants of missing items in the DAG model are missing as well. Hence, if this holds for all the records, then the resulting likelihood has the same functional form as the likelihood of complete data, i.e., it contains no summations. This leads to similar simplifications as the so-called monotone pattern of missingness described in Little and Rubin, 1987; see also Didelez and Pigeot, 1998. In the previous example, if X_3 was missing as well in record two, then the likelihood $\Pr(\boldsymbol{d}^1, \boldsymbol{o}^2|m, \boldsymbol{\theta})$ would be the product:

$$(1 - \theta_{X_1=1})^1 \theta^1_{X_2=1|X_1=0} \theta^1_{X_3=1|X_2=1} \cdot \theta^1_{X_1=1} \sum_{x_2, x_3} \theta_{x_3|x_2} \theta_{x_2|X_1=1} \quad =$$
$$(1 - \theta_{X_1=1})^1 \theta^1_{X_1=1} \theta^1_{X_2=1|X_1=0} \theta^1_{X_3=1|X_2=1}$$

Note that when learning the model of a BN, we in fact "build" (DAG) models by traversing the DAG search space. This means that the functional form of the likelihood will change as well. Hence, the above simplification in practice is of little use when learning models. Even when learning the parameter of BNs the pattern of missingness rarely is such that a significant reduction of summation terms can be achieved. One can attempt to reduce the number of terms by using an equivalent DAG model with alternative child/parents relations. For instance with the above data and the model $X_2 \rightarrow X_3$, the likelihood consists of a sum of two complete data likelihoods; for the equivalent model $X_3 \rightarrow X_2$ this can be reduced to a likelihood consisting of just one term. Unfortunately, for larger DAG models it is rarely possible to reduce the number of summation terms significantly.

The Bayesian approach is intimately related to the likelihood via Bayes' law, and may be seen as a kind of "wrapper" around the likelihood where the unobserved items are summed out. Starting with a product Dirichlet prior distribution, recall that the posterior is obtained as:

$$\Pr(\boldsymbol{\Theta}|m,\boldsymbol{o}) \propto \Pr(\boldsymbol{o}|m,\boldsymbol{\Theta})\Pr(\boldsymbol{\Theta}|m)$$

The likelihood of incomplete data is a sum of likelihoods of complete data. This means that the posterior parameter distribution $\Pr(\boldsymbol{\Theta}|m,\boldsymbol{o})$ becomes a *mixture distribution*; each component in the mixture is a product Dirichlet distribution corresponding to a completion of the data.

In the previous example, the mixture consists of two components, where each component is a product Dirichlet with normalising factors Z_1 and Z_2; here Z_i is the posterior normalising factor as function of the sufficient statistics from completion i as given in eq. 3.9. Z_i^{-1} is the normalising term for the product Dirichlet for completion i (the "mass under" the unnormalised product Dirichlet), and $\sum_j Z_j^{-1}$ is the normalising term for the entire unnormalised posterior (the "mass under" the sum of the unnormalised product Dirichlets). It follows that in order to normalise the resulting mixture as a whole, the mixture weights are determined as:

$$w_i = \frac{Z_i^{-1}}{\sum_j Z_j^{-1}} \Leftrightarrow w_i = \frac{\sum_j Z_j}{Z_i} \tag{5.3}$$

and the exact posterior is given by:

$$\Pr(\boldsymbol{\Theta}|m,\boldsymbol{o}) = \sum_j w_j \Pr(\boldsymbol{\Theta}|\boldsymbol{o},\boldsymbol{u}^{(j)},m)$$

Here $\boldsymbol{u}^{(j)}$ denotes the jth completion of the data. Hence, the mixture weights simply redistribute the mass according to the inverse of the normalising factors of the Dirichlets.

2.1.1 Complications for learning parameters

In order to learn the parameter of a BN the ML estimates can be used. For complete data these are given by eq. 3.4. With incomplete data, however, finding the parameter value that maximises the likelihood is no trivial task; we then have to turn to numerical optimisation techniques.

For the Bayesian approach to parameter learning the posterior mixture distribution may in itself be the solution. However, the huge number of mixture components makes it practically impossible to perform any

kind of statistical analysis, such as computing summary statistics. The main problem is not only that all possible completions actually have to be generated, but also that we in general need to retain the completions in memory. Although for a fixed model it suffices to save the sufficient statistics in memory and not the actual data completions, in general this is unmanageable and from a storage capacity point of view nearly impossible.

2.1.2 Bayesian sequential updating

For complete data, we saw that Bayesian updating per record yields the same result as batch updating. For incomplete data this is no longer the case (Spiegelhalter and Lauritzen, 1990; Cowell et al., 1995). In fact, for incomplete data, different orders results in different posteriors. The reason is that the Bayesian approach incorporates a predictive feature during the learning process, based on what has been gathered thus far. Once a record is observed where some items are missing, they are predicted using the available information that has been "learned" from the previous records. As an example, consider the incomplete data from the previous example. First, $\boldsymbol{d}^1$ is observed, resulting in the posterior $\Pr(\boldsymbol{\Theta}|m, \boldsymbol{d}^1)$, where conditioning on $\boldsymbol{d}^1$ means that the Dirichlet is updated with the counts extracted from record 1. Next, record two is considered, but this time there are items missing:

$$\Pr(\boldsymbol{\Theta}|m, \boldsymbol{d}^1, \boldsymbol{o}^2) = \sum_{\boldsymbol{u}^2} \Pr(\boldsymbol{\Theta}|\boldsymbol{d}^1, \boldsymbol{o}^2, \boldsymbol{u}^2) \Pr(\boldsymbol{u}^2|m, \boldsymbol{d}^1, \boldsymbol{o}^2)$$

where $\Pr(\boldsymbol{\Theta}|\boldsymbol{d}^1, \boldsymbol{o}^2, \boldsymbol{u}^2)$ is the posterior Dirichlet distribution updated with complete data from record 2, and $\Pr(\boldsymbol{u}^2|m, \boldsymbol{d}^1, \boldsymbol{o}^2)$ is the predictive distribution based on the posterior Dirichlet after seeing record 1, conditional on the items that actually have been observed in record 2. In this example X_2 is missing, and has two possible completions. The posterior $\Pr(\boldsymbol{\Theta}|m, \boldsymbol{d}^1, \boldsymbol{o}^2)$ becomes a mixture distribution consisting of two components, with mixing weights corresponding to the probability of completing the record with '1' and '0' based on what has been gathered from $\boldsymbol{d}^1$.

To compute the probability $\Pr(\boldsymbol{u}^2|m, \boldsymbol{d}^1, \boldsymbol{o}^2)$, inference in the BN is required where the parameter values are the expectation of the posterior Dirichlet $\Pr(\boldsymbol{\Theta}|m, \boldsymbol{d}^1)$, and evidence $\boldsymbol{o}^2$; this was discussed in Chapter 2, Section 5.1. After having processed record 2, the next time we are presented with an incomplete record, we require the expectation of a

mixture of two product Dirichlets. This is equal to the sum of the expectations of each Dirichlet component times their mixture weights.

This predictive behaviour goes on: the records are processed sequentially, and predictions are conditional on the records that have already been processed.

Suppose now that $\boldsymbol{d}^2$ and $\boldsymbol{d}^1$ are swapped. In that case the missing item X_2 needs to be predicted prior to seeing any other records. The prediction is based on the hyper parameter of the Dirichlet only. The next record does not contain any missing items, and once it has been observed, it follows that the posterior is a different one than had we been presented with the completely observed record first. The updating procedure just described is *in distribution*, i.e., the posterior parameter distribution is updated as new records are processed. The Dirichlet allows updating of the expectation only (Cowell, 1998). Hence, each new incoming record "moves" the expectation slightly without the need of maintaining the entire posterior mixture in memory. Still, the resulting posterior expectation depends on the order of the records.

Intuitively, the sensitivity to the ordering of the records is to be expected; after all not all of $\boldsymbol{o}$ is used for predictions, but only subsets thereof. In that sense, this sequential Bayesian way of updating does not fulfill our requirement of being a principled approach for analysing incomplete data. However, this does not mean that the approach is wrong as such. In fact, for dynamic systems where observations are sequentially produced as time passes, a Bayesian will only revise the "corpus" slightly when presented with a new observation. When presented with a batch of iid records however, the records are time-interchangeable, and the sequential approach strictly speaking does not apply.

2.1.3 Complications for learning models

Model learning methods rely on the likelihood function or the posterior parameter distribution (that in turn also depends on the likelihood function). The functional form of the likelihood for incomplete data makes it difficult to perform model learning.

In order to apply the penalised likelihood scoring criteria, the ML estimates are required. However, these are difficult to determine for incomplete data, hence the penalised likelihood scoring metrics can not be applied without further ado.

The Bayesian marginal likelihood is the normalising factor for the prior product Dirichlet over the normalising factor for the posterior prod-

uct Dirichlet in eq. 3.14. This derivation remains valid for incomplete data, with the exception that for incomplete data the denominator is then the normalising factor for the posterior *mixture* distribution. This normalising factor is only available given that the mixture is known, requiring that (almost) all completions are generated after which the normalisation factor for the posterior mixture can be computed as:

$$\frac{1}{\sum_j Z_j^{-1}}$$

where Z_j is the normalising factor for the posterior product Dirichlet as function of the sufficient statistics from the complete data $(\boldsymbol{o}, \boldsymbol{u}^{(j)})$, with $\boldsymbol{u}^{(j)}$ being the jth completion.

3. Principled iterative methods

Looking at the exact (batch) analysis, there is no "predictive" element; the unobserved items are summed out, without the interference by a predictive distribution (with exception to the Bayesian sequential approach). In a way this would be the "naive" approach that would come to mind as a first attempt to deal with incomplete data in a principled fashion: simply sum out the missing items, and use what is left (the observed part of the data) for further statistical analysis. As we sketched in the previous sections, this turns out to be a rather unwise approach.

In this section we discuss other principled algorithms for handling incomplete data problems. They provide an alternative to the exact methods discussed until now, and are from a computational point of view more tractable. The methods are iterative in nature, and converge towards (a) completion(s) of the incomplete data and consequently they are best interpreted as prediction-based approaches.

3.1 Expectation Maximisation—EM

In the following we investigate the well-known deterministic EM-algorithm by Dempster et al., 1977 for computing ML estimates. The EM-algorithm can be applied to a variety of problems that may conceptually be considered incomplete data problems or that can be formulated as such, e.g., for fitting mixture models. However, in the context of learning BN parameters, the problem formulation is straightforward. In Shafer, 1997; McLachlan and Krishnan, 1997; Lauritzen, 1995 the EM-algorithms is treated in more detail.

The EM-algorithm in contrast to the approaches discussed in section 2.1 employs a predictive distribution; in fact it makes use of several predictive distributions. The general idea is to predict missing items using the best predictive distribution that we are able to derive given what we know *at a given point in time*.

Such a predictive distribution is $\Pr(\boldsymbol{U}|m, \boldsymbol{o}, \boldsymbol{\theta}^{(t)})$, where $\boldsymbol{\theta}^{(t)}$ is the parameter based on information we have gathered up until now. This parametrisation enables us to produce the best predictions possible at time t.

The EM-algorithm departs from the likelihood decomposition:

$$\Pr(\boldsymbol{o}|m, \boldsymbol{\theta}) = \frac{\Pr(\boldsymbol{o}, \boldsymbol{U}|m, \boldsymbol{\theta})}{\Pr(\boldsymbol{U}|m, \boldsymbol{o}, \boldsymbol{\theta})} \tag{5.4}$$

Recall that the ML estimate is found by:

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \Pr(\boldsymbol{o}|m, \boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} \frac{\Pr(\boldsymbol{o}, \boldsymbol{U}|m, \boldsymbol{\theta})}{\Pr(\boldsymbol{U}|m, \boldsymbol{o}, \boldsymbol{\theta})}$$

Because the logarithm is a positive monotonic transformation, we may equivalently do the following:

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \log \Pr(\boldsymbol{o}, \boldsymbol{U}|m, \boldsymbol{\theta}) - \log \Pr(\boldsymbol{U}|m, \boldsymbol{o}, \boldsymbol{\theta}) \tag{5.5}$$

Hence, taking the logarithm will not have any influence on the estimation of $\boldsymbol{\theta}$. In the limit EM returns $\hat{\boldsymbol{\theta}}$ essentially solving eq. 5.5.

The missing values are predicted by computing the expectation of (the logarithm of) the likelihood given in eq. 5.4. This results in the observed log-likelihood:

$$\begin{aligned} \sum_{\boldsymbol{u}} \Big(\log \Pr(\boldsymbol{o}|m, \boldsymbol{\theta}) \Big) \Pr(\boldsymbol{u}|m, \boldsymbol{o}, \boldsymbol{\theta}^{(t)}) &= \log \Pr(\boldsymbol{o}|m, \boldsymbol{\theta}) \\ &= Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)}) - H(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)}) \end{aligned}$$

where:

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)}) = \sum_{\boldsymbol{u}} \Big(\log \Pr(\boldsymbol{o}, \boldsymbol{u}|m, \boldsymbol{\theta}) \Big) \Pr(\boldsymbol{u}|m, \boldsymbol{o}, \boldsymbol{\theta}^{(t)})$$

$$H(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)}) = \sum_{\boldsymbol{u}} \Big(\log \Pr(\boldsymbol{u}|m, \boldsymbol{o}, \boldsymbol{\theta}) \Big) \Pr(\boldsymbol{u}|m, \boldsymbol{o}, \boldsymbol{\theta}^{(t)})$$

Let $\boldsymbol{\theta}^{(t+1)}$ be the value of $\boldsymbol{\theta}$ that maximises $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)})$. $\boldsymbol{\theta}^{(t+1)}$ is then a better estimate than $\boldsymbol{\theta}^{(t)}$ because $\Pr(\boldsymbol{o}|m, \boldsymbol{\theta}^{(t+1)}) - \Pr(\boldsymbol{o}|m, \boldsymbol{\theta}^{(t)}) \geq 0$. This can be seen from the following:

$$Q(\boldsymbol{\theta}^{(t+1)}, \boldsymbol{\theta}^{(t)}) - Q(\boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}^{(t)}) + H(\boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}^{(t)}) - H(\boldsymbol{\theta}^{(t+1)}, \boldsymbol{\theta}^{(t)}) \geq 0$$

In this equation $Q(\boldsymbol{\theta}^{(t+1)}, \boldsymbol{\theta}^{(t)}) \geq Q(\boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}^{(t)})$ because $\boldsymbol{\theta}^{(t+1)}$ was chosen to accomplish exactly that. Notice that actually we need not even maximise $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)})$, but we need only choose a $\boldsymbol{\theta}^{(t+1)}$ that yields a larger Q-value than $\boldsymbol{\theta}^{(t)}$. Also $H(\boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}^{(t)}) \geq H(\boldsymbol{\theta}^{(t+1)}, \boldsymbol{\theta}^{(t)})$ since:

$$\begin{aligned}
H(\boldsymbol{\theta}^{(t+1)}, \boldsymbol{\theta}^{(t)}) - H(\boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}^{(t)}) &= \mathrm{E}[\log \frac{\Pr(\boldsymbol{U}|m, \boldsymbol{o}, \boldsymbol{\theta}^{(t+1)})}{\Pr(\boldsymbol{U}|m, \boldsymbol{o}, \boldsymbol{\theta}^{(t)})} | \boldsymbol{\theta}^{(t)}] \\
&\leq \log \mathrm{E}[\frac{\Pr(\boldsymbol{U}|m, \boldsymbol{o}, \boldsymbol{\theta}^{(t+1)})}{\Pr(\boldsymbol{U}|m, \boldsymbol{o}, \boldsymbol{\theta}^{(t)})} | \boldsymbol{\theta}^{(t)}] \\
&= \log \sum_{\boldsymbol{u}} \frac{\Pr(\boldsymbol{u}|m, \boldsymbol{o}, \boldsymbol{\theta}^{(t+1)})}{\Pr(\boldsymbol{u}|m, \boldsymbol{o}, \boldsymbol{\theta}^{(t)})} \Pr(\boldsymbol{u}|m, \boldsymbol{o}, \boldsymbol{\theta}^{(t)}) \\
&= \log 1 = 0
\end{aligned}$$

Here Jensen's Inequality $\mathrm{E}[\log(\cdot)] \leq \log \mathrm{E}[\cdot]$ with the concave log-function has been applied.

The key lies in the Q-function since only that function is relevant for selecting $\boldsymbol{\theta}^{(t+1)}$; the H-function we may leave out of the picture. The Q-function is defined as the expectation of $\log \Pr(\boldsymbol{o}, \boldsymbol{U}|m, \boldsymbol{\theta})$—the log-likelihood of the complete data. Because the multinomial distribution belongs to the regular exponential family, the log-likelihood is linear in the sufficient statistics. Exploiting linearity of expectation, this means that the expectation of the log-likelihood reduces to the expectation of the sufficient statistics. Due to the iid assumption the expected sufficient statistics $\mathrm{E}[n(x_i, \boldsymbol{x}_{pa(i)})]$ with respect to the predictive distribution become:

$$\mathrm{E}[n(x_i, \boldsymbol{x}_{pa(i)})|\boldsymbol{\theta}^{(t)}] = \sum_{j=1}^{c} \Pr(X_i = x_i, \boldsymbol{X}_{pa(i)} = \boldsymbol{x}_{pa(i)}|\boldsymbol{o}^j, m, \boldsymbol{\theta}^{(t)})$$

If X_i and $\boldsymbol{X}_{pa(i)}$ are observed in record j, and have values x_i and $\boldsymbol{x}_{pa(i)}$ then $\Pr(x_i, \boldsymbol{x}_{pa(i)}|\boldsymbol{o}^j, m, \boldsymbol{\theta}^{(t)}) = 1$ (and equal to 0 for any other value of X_i and $\boldsymbol{X}_{pa(i)}$), and inference in the BN is not required. Otherwise the result of the inference is added to the sufficient statistics. It is now easy

to find the parameter $\boldsymbol{\theta}^{(t+1)}$; it is the ML estimate using the expected sufficient statistics.

The EM-algorithm is usually presented using the likelihood approach the way just described. It is not difficult to prove that EM also holds in a Bayesian setting. The only essential difference is that the parameter estimates are based not only on the data, but also on the hyper parameter $\boldsymbol{\alpha}$.

In summary, the EM-algorithm consists of two steps:

---

1 E-step: Predict missing values given the current best estimate of $\boldsymbol{\theta}$. In the case of a multinomial sample, this corresponds to predicting the cell counts.

2 M-step: Calculate the parameter estimate using the statistics from the E-step, and consider it the new best estimate of $\boldsymbol{\theta}$.

Repeated application of the above E- and M-step produces a sequence of statistics, i.e., complete data samples, and parameters. The parameters will generally converge towards the true parameter of the BN given the observed data $\boldsymbol{o}$. The rate of convergence is related to the so-called *fraction of missing information* in the data sample. Usually it will take longer to converge if more data is missing (McLachlan and Krishnan, 1997), but not necessarily so. The fraction of missing information need not change dramatically for different fractions of missing items.

The rate of convergence of course also depends on the initial estimate at time $t = 0$. If a qualified guess is available, it can be used as a plausible starting point. If the likelihood function is multimodal and the curvature contains regions with ridges, convergence may be slow, and local maxima may be found. To diagnose problems like these several different starting points should be tried.

EM converges monotonically to the (local) optimal parameter estimate. We may monitor its progress by studying the difference between the estimated parameter values between each iteration. If the absolute difference is below a certain threshold it means that we may stop the algorithm, and consider the last estimate as the required parameter estimate given $\boldsymbol{o}$.

The idea behind EM is quite appealing, and easy to understand. The application of the E-step however includes an inference step, which for

BNs is known to be an NP-hard problem (Cooper, 1990). The inference is performed on a per record basis, and repeated iteration of the E- and M-step means that we potentially have to perform many inference calculations before convergence.

3.1.1 Structural EM—SEM

EM is traditionally used for parameter learning, however, it is possible to use the statistics obtained from the E-step for model selection as well. Hence, once EM has converged, the statistics corresponding to completions of the incomplete data can be used together with the complete data model scoring metrics discussed in chapter 3. Unfortunately it is computationally expensive to use EM for model selection in this way because even a minor change in model structure requires a complete re-run of the EM-algorithm for the new structure.

It is in fact possible to do model selection *within* EM such that the best model is selected in the limit. This algorithm is referred to as *Structural Expectation Maximisation* (SEM), originally proposed by Friedman, 1997; Friedman, 1998. A slightly inefficient forerunner of the SEM-algorithm was proposed by Singh, 1997.

In Friedman, 1997 SEM was suggested and proved based on the penalised likelihood model scoring metric, MDL/BIC. In Friedman, 1998 the marginal likelihood scoring metric was used instead, for which SEM was shown to be at least approximately correct.

PROPOSITION 5.1 SEM *can be made exact in combination with the marginal likelihood score, and it converges to (a local) maximum of* $\Pr(M|\boldsymbol{o})$.

We exploit the fact that for model selection the marginal likelihood can be rewritten as a penalised likelihood score; it is basically just another penalised likelihood model scoring metric.

We seek the most probable BN given $\boldsymbol{o}$, i.e., we seek the MAP of:

$$\Pr(M, \boldsymbol{\Theta}|\boldsymbol{o}) = \Pr(\boldsymbol{\Theta}|M, \boldsymbol{o}) \Pr(M|\boldsymbol{o})$$

From Chapter 3, Section 3.3, we know that we may equivalently maximise the penalised log-likelihood:

$$\log \Pr(\boldsymbol{o}, \boldsymbol{\alpha} - \mathbf{1}|M, \boldsymbol{\Theta}) + \log f(M, \boldsymbol{\alpha})$$

Similar to the EM-algorithm, we introduce the predictive distributions $\Pr(\boldsymbol{U}|\boldsymbol{o}, (m, \boldsymbol{\theta})^{(t)})$, but now the model *and* the parameter are dependent on time t.

Following the same line of reasoning as we did for the EM derivation, we may restrict attention to the following Q-function, $Q((m,\boldsymbol{\theta}),(m,\boldsymbol{\theta})^{(t)})$:

$$\sum_{\boldsymbol{u}} \Big(\log \Pr(\boldsymbol{o},\boldsymbol{u},\boldsymbol{\alpha}-\mathbf{1}|m,\boldsymbol{\theta}) + \log f(m,\boldsymbol{\alpha}) \Big) \Pr(\boldsymbol{u}|\boldsymbol{o},(m,\boldsymbol{\theta})^{(t)})$$

As before, the log-likelihood is linear in the sufficient statistics from the data, and by adding the logarithm of the penalty term, the whole expression remains a linear function of the sufficient statistics. As with EM, this means that the expectation of the entire expression reduces to the expectation of the sufficient statistics plus the penalty term.

Using the same argument as for EM, it follows that by letting $(m,\boldsymbol{\theta})^{(t+1)}$ be the value of $(m,\boldsymbol{\theta})$ that maximises $Q((m,\boldsymbol{\theta}),(m,\boldsymbol{\theta})^{(t)})$, it is a better estimate than $(m,\boldsymbol{\theta})^{(t)}$.

It is not difficult to find $(m,\boldsymbol{\theta})^{(t+1)}$, because we know that for any given m, $\log \Pr(\boldsymbol{o},\boldsymbol{u},\boldsymbol{\alpha}-1|m,\boldsymbol{\theta}) + \log f(m,\boldsymbol{\alpha})$ is maximised by the ML estimate $\hat{\boldsymbol{\theta}}$ based on the statistics from $(\boldsymbol{o},\boldsymbol{u},\boldsymbol{\alpha}-\mathbf{1})$. Thus, we select a model $m^{(t+1)} = \hat{m}$ that maximises this penalised log-likelihood using $\boldsymbol{\theta}^{(t+1)} = \hat{\boldsymbol{\theta}}$, and then use the pair $(\hat{m},\hat{\boldsymbol{\theta}})^{(t+1)}$ in the predictive distribution for the next iteration.

The SEM-algorithm thus consists of the following steps:

---

1 E-step: Predict missing values in terms of the cell counts given the current estimate of $(m,\boldsymbol{\theta})$.

2 M-step: Select the new best pair $(m,\boldsymbol{\theta})$:

   (a) Using the statistics from the E-step together with the marginal likelihood scoring criterion, select the best model $\hat{m}$.

   (b) Calculate the parameter estimate $\hat{\boldsymbol{\theta}}$ for $\hat{m}$ using the statistics from the E-step.

So how is this different from the naive model selection approach in EM? Instead of using the same model for the predictive distribution at time $t+1$, the model is changed such that it captures the independences of the completed data at time t. In the naive approach there are too many wasteful iterations of fitting the parameter to models that are far from being correct.

3.2 Data Augmentation—DA

In the next few sections, we work towards an MCMC model sampler, that produces realisations from $\Pr(M|\boldsymbol{o})$. We introduce several intermediate MCMC samplers on our way. We take a Bayesian approach, and treat both the parameter and the model as random variables. A clear distinction between the model learning phase and the parameter learning phase is in the Bayesian approach a bit awkward. In the following we therefore can't leave $\boldsymbol{\Theta}$ entirely out of the picture even though our ultimate goal is to learn about models only. We depart from an MCMC sampler called Data Augmentation (DA).

DA is an MCMC Gibbs sampler that resembles EM in many ways. DA fits perfectly into the Bayesian paradigm in the sense that it accounts for the *uncertainty due to missing data*; the increased variance that missing data introduces. It does so by returning realisations from distributions, rather than returning a single summary statistic like EM does. As the name suggests, DA *augments* the incomplete data, i.e., it augments $\boldsymbol{o}$ with $\boldsymbol{U}^{(t)}$ thus creating complete data sets. In the standard form, DA as a side-effect also creates the corresponding parameters $\boldsymbol{\Theta}^{(t)}$.

It should be noted that although we in this section reduce DA to a special kind of Gibbs sampling, the original presentation and motivation by Tanner and Wong, 1987 was slightly different.

Consider the joint distribution $\Pr(\boldsymbol{U}, \boldsymbol{\Theta}|m, \boldsymbol{o})$, i.e., the joint distribution of missing data and parameter $\boldsymbol{\Theta}$. Recall that Gibbs sampling produces draws from a joint distribution, given that we are able to draw from the full conditionals. In this case this means that we may obtain realisations from the above joint, if we can draw from the distributions $\Pr(\boldsymbol{U}|m, \boldsymbol{\Theta}, \boldsymbol{o})$ and $\Pr(\boldsymbol{\Theta}|m, \boldsymbol{o}, \boldsymbol{U})$.

Consider now the following Gibbs sampling set-up:

$$\begin{aligned} \boldsymbol{U}^{(t+1)} &\sim \Pr(\boldsymbol{U}|m, \boldsymbol{\theta}^{(t)}, \boldsymbol{o}) \\ \boldsymbol{\Theta}^{(t+1)} &\sim \Pr(\boldsymbol{\Theta}|m, \boldsymbol{o}, \boldsymbol{u}^{(t+1)}) \\ &\vdots \end{aligned}$$

For $t \to \infty$, the realisations come from the invariant joint distribution $\Pr(\boldsymbol{U}, \boldsymbol{\Theta}|m, \boldsymbol{o})$. If we restrict attention to the realisations of $\boldsymbol{U}$ or $\boldsymbol{\Theta}$ on their own, we may consider those as coming from the predictive distribution $\Pr(\boldsymbol{U}|m, \boldsymbol{o})$ or the posterior parameter distribution $\Pr(\boldsymbol{\Theta}|m, \boldsymbol{o})$.

Using the theory presented in Chapter 4, we can now for instance compute the empirical average of some function over the n augmented

data sets:

$$\mathrm{E}[h(\boldsymbol{U},\boldsymbol{o})] = \sum_{\boldsymbol{u}} h(\boldsymbol{u},\boldsymbol{o}) \Pr(\boldsymbol{u}|m,\boldsymbol{o}) \approx \frac{1}{n}\sum_{t=1}^{n} h(\boldsymbol{u}^{(t)},\boldsymbol{o}) \tag{5.6}$$

For example $h(\cdot)$ may be the function that counts the sufficient statistics. The expression in eq. 5.6 will in that case give us the expected sufficient statistics. These statistics can then be used as input for some other algorithm. If the initial realisations of $\boldsymbol{U}$ are not from the invariant distribution, it is a good idea to skip those samples when computing the empirical average in eq. 5.6 (although this is not a requirement because for (very) large n the burn-in has a negligible effect on the approximation anyway).

DA in BNs is reasonably easy to do thanks to the BN decomposition. In this regard the Markov blanket plays an important role. If we focus on one single record, we require samples from $\Pr(\boldsymbol{U}^l, \boldsymbol{\Theta}|\boldsymbol{o}^l, m)$. If we create the augmented BN as we did in figure 2.4 in Chapter 2, Section 5.1, we see that if we apply a single-site Gibbs sampler, we may for each variable restrict attention to its Markov blanket, no matter if it is a parameter variable or a variable in $\boldsymbol{U}^l$.

In our context we have c records, and drawing from $\Pr(\boldsymbol{U}|m,\boldsymbol{\Theta}^{(t)},\boldsymbol{o})$ in the Gibbs sampler means drawing a large multivariate sample. Exploiting the iid assumption, each record with missing items may be considered separately, i.e., for each record we draw $\boldsymbol{U}^l$ given $\boldsymbol{o}^l$ and the current $\boldsymbol{\Theta}^{(t)} = \boldsymbol{\theta}$. Draw $\boldsymbol{U}^l$ using the following univariate Gibbs sampler:

$$\begin{aligned}
U_1^{l,(t+1)} &\sim \Pr(U_1^l|m, u_2^{l,(t)}, \ldots, u_{r(l)}^{l,(t)}, \boldsymbol{o}^l, \boldsymbol{\theta}) \\
&\vdots \\
U_{r(l)}^{l,(t+1)} &\sim \Pr(U_{r(l)}^l|m, u_1^{l,(t+1)}, \ldots, u_{r(l)-1}^{l,(t+1)}, \boldsymbol{o}^l, \boldsymbol{\theta}) \\
&\vdots
\end{aligned}$$

In the limit this process generates samples from $\Pr(\boldsymbol{U}^l|m,\boldsymbol{\theta},\boldsymbol{o}^l)$, i.e., realisations from the predictive distribution for record l, given $\boldsymbol{\theta}$. For drawing U_i from the full conditional it suffices to only include those variables on the conditional side that are part of the Markov blanket of U_i. Observe that we actually apply a sub-MCMC Gibbs sampler when we need to draw from $\Pr(\boldsymbol{U}|m,\boldsymbol{\Theta}^{(t)},\boldsymbol{o})$; hence we should not expect

the initial full completions to be multivariate samples from the desired target distribution.

Once a $\boldsymbol{U}^{(t+1)}$ has been generated, realisations from $\Pr(\boldsymbol{\Theta}|m, \boldsymbol{o}, \boldsymbol{U}^{(t+1)})$ are required, which is now a distribution conditional on complete data. Because of parameter independence, we may draw the parameter vectors $\boldsymbol{\Theta}_{X_i|\boldsymbol{x}_{pa(i)}}$ from the posterior Dirichlet $\Pr(\boldsymbol{\Theta}_{X_i|\boldsymbol{x}_{pa(i)}}|m, \boldsymbol{o}, \boldsymbol{U}^{(t+1)})$ independently of each other, i.e., for all X_i and each parent set configuration $\boldsymbol{x}_{pa(i)}$.

Sampling from the Dirichlet amounts to drawing from the *Gamma distributions* (Gelman et al., 2004). Given parent set configuration $\boldsymbol{x}_{pa(i)}$, for each value x_i, draw $v(x_i)$ from the Gamma distribution with shape parameter $s(x_i, \boldsymbol{x}_{pa(i)}) = \alpha(x_i, \boldsymbol{x}_{pa(i)}) + n(x_i, \boldsymbol{x}_{pa(i)})$ and common scale parameter (e.g. 1). Let $\theta_{x_i|\boldsymbol{x}_{pa(i)}} = v(x_i)/\sum_{x_i} v(x_i)$ be the normalised elements of $\boldsymbol{\theta}_{X_i|\boldsymbol{x}_{pa(i)}}$, then this vector will have a Dirichlet distribution $\boldsymbol{\Theta}_{X_i|\boldsymbol{x}_{pa(i)}} \sim \mathrm{Dir}(\boldsymbol{\Theta}_{X_i|\boldsymbol{x}_{pa(i)}}|\boldsymbol{s})$. The required steps for drawing from the Gamma distribution are well-described in literature; see for instance Kennedy and Gentle, 1980.

In summary, DA consists of the following two steps:

---

1 I-step: The *imputation* step predicts missing values by drawing from $\Pr(\boldsymbol{U}|m, \boldsymbol{\theta}^{(t)}, \boldsymbol{o})$, where $\boldsymbol{\theta}^{(t)}$ was drawn from the last parameter posterior.

2 P-step: The *posterior* step draws the BN parameter from the posterior $\Pr(\boldsymbol{\Theta}|m, \boldsymbol{o}, \boldsymbol{U}^{(t+1)})$, where $\boldsymbol{U}^{(t+1)}$ was the prediction drawn for the missing values from the last I-step.

The strong resemblance with EM is evident. The I-step corresponds to the E-step in EM and the P-step corresponds to the M-step. The more fundamental difference with EM, is that DA produces *realisations* from $\Pr(\boldsymbol{U}, \boldsymbol{\Theta}|m, \boldsymbol{o})$, rather than just returning the most probable parameter and corresponding completion of the data. In that sense DA is a Bayesian approach because entire distributions are approximated via empirical samples.

The chain of realisations evolving by repeated application of the I-step is also the key to *multiple imputation* (MI) (Little and Rubin, 1987; Rubin, 1987). MI is a technique that combines in a non-Bayesian way

several of the data completions such that the increased uncertainty due to missing data is accounted for, when further statistical analysis of incomplete data is necessary. It does so by combining the so-called *in-between variance* of independent completions obtained from the I-step of DA. Details can be found in for instance Brand, 1999.

3.2.1 DA and eliminating the P-step—DA-P

From a predictive point of view, the goal is to complement the incomplete data sample; with DA we get several probable completions, such that we can average any function over all those completions as in eq. 5.6. In this respect only the I-step is interesting, since this is where the realisations from the predictive distribution are generated. The realisations of the parameter in the P-step are not directly relevant for our purpose, but the P-step is necessary for standard DA to work. As an alternative to standard DA we suggest the following alternative Gibbs sampler (Riggelsen, 2004; Riggelsen, 2006b), where the P-step is omitted effectively eliminating the overhead of having to draw from the product Dirichlet distribution. Additionally, the Markov chain converges faster to $\Pr(\boldsymbol{U}|m,\boldsymbol{o})$ than to the joint $\Pr(\boldsymbol{U},\boldsymbol{\Theta}|m,\boldsymbol{o})$ because there is one dimension less "to explore".

First off, we require the following:

PROPOSITION 5.2 *For a completion* $\boldsymbol{U}^{(t)}$*, we have* $\Pr(\boldsymbol{U}|\boldsymbol{U}^{(t)},\boldsymbol{o},m) \propto \Pr(\boldsymbol{U}|\boldsymbol{o},(\boldsymbol{U}^{(t)},\boldsymbol{o}),m) = \Pr(\boldsymbol{U}|\boldsymbol{o},\boldsymbol{\theta} = \mathrm{E}[\boldsymbol{\Theta}|\boldsymbol{U}^{(t)},\boldsymbol{o}],m)$ *where in the last distribution* $\boldsymbol{o}$ *is evidence and* $(\boldsymbol{\theta} = \mathrm{E}[\boldsymbol{\Theta}|(\boldsymbol{U}^{(t)},\boldsymbol{o})],m)$ *a BN.*

The equivalence between the 1st and the 2nd equation is up to the constant factor $1/\Pr(\boldsymbol{o}|m)$. The equality follows from the predictive property of Bayesian statistics viz. eq. 2.5 after seeing $\boldsymbol{d}$, a whole set of completely observed records:

$$\begin{aligned}\Pr(\boldsymbol{X}|\boldsymbol{d},m) &= \int \prod_{i=1}^{p} \Pr(X_i|\boldsymbol{X}_{pa(i)},\boldsymbol{d},\boldsymbol{\theta},m)\Pr(\boldsymbol{\theta}|\boldsymbol{d},m)d\boldsymbol{\theta} \\ &= \Pr(\boldsymbol{X}|\boldsymbol{\theta} = \mathrm{E}[\boldsymbol{\Theta}|\boldsymbol{d}],m)\end{aligned}$$

In particular, for record l we have $\boldsymbol{D}^l = (\boldsymbol{O}^l,\boldsymbol{U}^l)$, we consider $\boldsymbol{O}^l$ as the evidence, and we may write:

$$\Pr(\boldsymbol{U}^l|\boldsymbol{o}^l,\boldsymbol{d},m) = \Pr(\boldsymbol{U}^l|\boldsymbol{o}^l,\boldsymbol{\theta} = \mathrm{E}[\boldsymbol{\Theta}|\boldsymbol{d}],m)$$

Consider now the following Gibbs sampler. Draw the $t+1$th imputation:

$$\begin{aligned} \boldsymbol{U}^{(t+1)} &\sim \Pr(\boldsymbol{U}|\boldsymbol{o}, \boldsymbol{u}^{(t)}, m) \propto \Pr(\boldsymbol{U}|\boldsymbol{o}, \boldsymbol{\theta} = \mathrm{E}[\boldsymbol{\Theta}|\boldsymbol{o}, \boldsymbol{u}^{(t)}], m) \\ &\vdots \end{aligned}$$

Thus a new imputation is drawn at time $t+1$ based on the last completed data sample at time t. In the limit we obtain realisations from the predictive distribution, $\Pr(\boldsymbol{U}|m, \boldsymbol{o})$. We refer to this sampler as DA-P (Data Augmentation minus the P-step).

The proposed sampling procedure is almost similar to normal DA. However, instead of *drawing* from the posterior parameter distribution as one would do in DA (the P-step of DA), an *expectation* step can be performed instead as a consequence of the Bayesian way of marginalising out the parameter space. The Markov chain does not converge to $\Pr(\boldsymbol{U}, \boldsymbol{\Theta}|m, \boldsymbol{o})$ but to $\Pr(\boldsymbol{U}|m, \boldsymbol{o})$, and the parameter space is not explored by the MCMC sampler. We thus only need to focus on the so-called *primary* chain (or the *chain of interest*) if we are interested in completions of the incomplete data. In the next section on learning models, this is exactly the chain we are interested in.

Although the parameter realisations are not explicitly drawn anymore as in DA, we may (for instance afterwards) approximate the posterior parameter distribution using the empirical average based on the completions drawn in the I-step of DA-P:

$$\Pr(\boldsymbol{\Theta}|m, \boldsymbol{o}) = \sum_{\boldsymbol{u}} \Pr(\boldsymbol{\Theta}|m, \boldsymbol{o}, \boldsymbol{u}) \Pr(\boldsymbol{u}|m, \boldsymbol{o}) \approx \frac{1}{n} \sum_{t=1}^{n} \Pr(\boldsymbol{\Theta}|m, \boldsymbol{o}, \boldsymbol{u}^{(t)})$$

If we instead of uniform mixture weights, $1/n$, assign weights according to eq. 5.3, we obtain a better approximation, which may be regarded as a finite mixture distribution with product Dirichlet components. We may now analyse the approximation as a real mixture distribution and compute summary statistics; for instance the variance may be computed giving an indication of the uncertainty, including uncertainty due to missing data (Riggelsen, 2006b); see Titterington et al., 1985 for more on statistical analysis of finite mixtures.

THEOREM 5.1 *The DA-P sampler produces an ergodic Markov chain with invariant distribution* $\Pr(\boldsymbol{U}|\boldsymbol{o}, m)$.

We need to make sure that the chain is irreducible, and this we do by checking the positivity requirement. We draw from $\Pr(\boldsymbol{U}|\boldsymbol{o}, \boldsymbol{\theta} =$

$\mathrm{E}[\boldsymbol{\Theta}|\boldsymbol{o}, \boldsymbol{U}^{(t)}], m)$, which indeed is strictly positive; there are no logically impossible combinations of $\boldsymbol{X}$ because from eq. 3.10 we see that $\boldsymbol{\theta} > \mathbf{0}$, and the BN is just some product of the elements from $\boldsymbol{\theta}$. Aperiodicity holds because there is a probability > 0 of drawing the same configuration as the last one, no matter the restrictions imposed by the model.

3.3 DA-P and model learning—MDA-P

When applying DA-P, we assume that the model is fixed indicated by the lower-case m on the conditional side. We now turn to the problem of finding models that can explain the data, but in a way that uncertainty due to missing data is taken into account when learning. The approach is somewhat similar to the deterministic SEM-algorithm.

The sampler is built around a Gibbs sampler, where samples in the limit come from the joint distribution $\Pr(\boldsymbol{U}, M|\boldsymbol{o})$. We omit the P-step, and draw realisations as follows:

$$\begin{aligned} \boldsymbol{U}^{(t+1)} &\sim \Pr(\boldsymbol{U}|\boldsymbol{o}, m^{(t)}, \boldsymbol{u}^{(t)}) \propto \Pr(\boldsymbol{U}|m^{(t)}, \boldsymbol{\theta} = \mathrm{E}[\boldsymbol{\Theta}|\boldsymbol{o}, \boldsymbol{u}^{(t)}], \boldsymbol{o}) \\ M^{(t+1)} &\sim \Pr(M|\boldsymbol{o}, \boldsymbol{u}^{(t+1)}) \\ &\vdots \end{aligned}$$

In the limit, the models thus obtained can be regarded as realisations from the posterior model distribution, $\Pr(M|\boldsymbol{o})$. We see that the completions drawn using DA-P augment the incomplete part of the data, and the now complete sample is used for model learning.

In Section 3.2.1 we already discussed how to draw from the distribution $\Pr(\boldsymbol{U}|M, \mathrm{E}[\boldsymbol{\Theta}|\boldsymbol{o}, \boldsymbol{U}^{(t)}], \boldsymbol{o})$, by filling in the unobserved part of the data sample on a per record basis. However, drawing from $\Pr(M|\boldsymbol{o}, \boldsymbol{U}^{(t+1)})$ is more complicated, and requires the application of a separate MCMC sampler. In Chapter 3, Section 3, an efficient MCMC model sampler was presented, called MB-MCMC. This means that when we need draws from the $\Pr(M|\boldsymbol{o}, \boldsymbol{U}^{(t+1)})$, we call the model sampler, which we know has $\Pr(M|\boldsymbol{o}, \boldsymbol{U}^{(t+1)})$ as the invariant distribution.

In summary, the MDA-P sampler (Model Data Augmentation minus the P-step) goes as follows:

---

1 I-step: The *imputation* step predicts missing values by drawing from $\Pr(\boldsymbol{U}|m^{(t)}, \mathrm{E}[\boldsymbol{\Theta}|\boldsymbol{o}, \boldsymbol{u}^{(t)}], \boldsymbol{o})$, where $\boldsymbol{u}^{(t)}$ was drawn from the last I-step, and $m^{(t)}$ was drawn from last the model posterior.

2 MP-step: The *model posterior* step draws from $\Pr(M|\boldsymbol{o}, \boldsymbol{u}^{(t+1)})$, where $\boldsymbol{u}^{(t+1)}$ was drawn from the last I-step.

THEOREM 5.2 *The MDA-P sampler produces an ergodic Markov chain with invariant distribution* $\Pr(M|\boldsymbol{o})$.

MDA-P combines two MCMC samplers. Provided that both samplers produce ergodic chains, the desired joint distribution of the overall sampler is reached in the limit (Tierney, 1994). We know that DA-P is ergodic with the target distribution $\Pr(\boldsymbol{U}|M^{(t)}, \mathrm{E}[\boldsymbol{\Theta}|\boldsymbol{o}, \boldsymbol{U}^{(t)}], \boldsymbol{o})$ and the same holds for the model sampler which has the target distribution $\Pr(M|\boldsymbol{o}, \boldsymbol{U}^{(t+1)})$.

If we compare MDA-P to the SEM-algorithm, then the I-step would correspond to the E-step in SEM, and the MP-step would correspond to the model selection step.

3.4 Efficiency issues of MDA-P

The efficiency of the MDA-P sampler presented in the previous section, depends on several factors. Although MDA-P provably will produce draws from the model posterior in the limit, this is only a theoretical guarantee, and the remarks made about MCMC sampling in Chapter 4 apply. Essentially we are dealing with a non-trivial MCMC sampler, and in the following two sections we investigate the sampler focusing on efficiency issues.

3.4.1 Properties of the sub-MCMC samplers

In the MDA-P sampler, we have marginalised out the parameter space such that the Gibbs sampler need not waste time in exploring this dimension. Moreover, as a side-effect of this, we need not implement a procedure for drawing from the product Dirichlet distribution. What remains is to draw imputations in the I-step, and generating models in the MP-step. Since both of these steps employ MCMC samplers on their own, the performance of MDA-P largely depends on the properties of those sub-MCMC samplers.

By calling the MB-MCMC algorithm in the MP-step, we made an attempt to perform reasonably well (in terms of mixing) in this sub-MCMC sampler by way of blocking edges. However, this does not mean

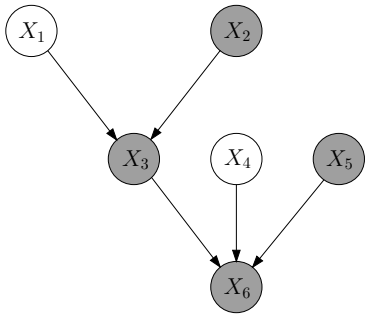

Figure 5.1. Gibbs sample from $\Pr(X_1, X_4|m, \boldsymbol{\theta}, X_2, X_3, X_5, X_6)$. The shaded vertices are observed, and block influence between X_1 and X_4 effectively improving the mixing properties of the Gibbs sampler.

that this step is easy. From a computational point of view, model sampling via MCMC is quite expensive.

The I-step consists of filling-in the unobserved part of the data. This step calls a rather simple Gibbs sampler using a univariate Gibbs sampler on a per record basis. The performance properties of this sub-MCMC sampler depends heavily on the fraction of missing data, and also on the model M at that point in time.

Recall from Section 3.2 that producing realisations in the I-step amounts to drawing from a Gibbs sampler, where each draw is from a univariate distribution with on the conditional side the variables of the Markov blanket and a fixed parameter. When missing items never occur on the conditional side at any time in the process of drawing multivariate realisations, then the convergence is immediate; the variables for which no values have been observed, are fully separated from each other given the remaining observed variables in the Markov blanket.

As an example consider the model m in figure 5.1. Suppose that for a given record, the shaded vertices are observed, and that X_1 and X_4 are missing. We require realisations from $\Pr(X_1, X_4|m, \boldsymbol{\theta}, X_2, X_3, X_5, X_6)$. Using the univariate Gibbs sampler we draw according to the following procedure:

$$
\begin{aligned}
X_1^{(t+1)} &\sim \Pr(X_1|X_2, X_3, x_4^{(t)}, X_5, X_6, m, \boldsymbol{\theta}) = \Pr(X_1|X_2, X_3, m, \boldsymbol{\theta}) \\
X_4^{(t+1)} &\sim \Pr(X_4|x_1^{(t+1)}, X_2, X_3, X_5, X_6, m, \boldsymbol{\theta}) = \Pr(X_4|X_3, X_5, X_6, m, \boldsymbol{\theta}) \\
&\vdots
\end{aligned}
$$

Because the shaded vertices form the Markov blanket of X_1 and X_4, the two variables are independent of each other. The Gibbs sampler reaches the invariant distribution in one single iteration. This means that the pair $(x_1^{(t+1)}, x_4^{(t+1)})$ is a sample from $\Pr(X_1, X_4|X_2, X_3, X_5, X_6, m, \boldsymbol{\theta})$. In fact we have a perfectly mixing chain, since samples from t to $t+1$ are iid.

From the above discussion it follows that for records with only few missing items, there are less interaction effects between variables with missing observations, and Gibbs sampling performs well. For records with many missing values, there are potentially many more interactions, and as a result the Gibbs sampler needs a longer burn-in period, and may have poor mixing properties.

The performance of the sampler is also dependent on the model, because it is the model that determines the Markov blanket, and thus the way variables depend on each other. A sparse DAG model has less interactions that a more dense DAG model, and therefore the Gibbs sampler performs better with the former than the latter.

In summary, the I-step may seem quite easy from an operational point of view, but from a computational point of view producing imputations via MCMC may be rather expensive.

We may conclude that each of the sub-MCMC samplers may be potentially problematic in terms of efficiency.

3.4.2 Interdependence between samplers

The performance of each of the two sub-MCMC samplers is important, but the dependence between the two sub-MCMC samplers also plays a significant role in the performance of MDA-P; afterall the two sub-MCMC samplers are conditional on the output of each other.

Spending too much time at each sub-MCMC sampler is potentially worthless if the overall chain is far from the invariant distribution; there is no need to let either sub-MCMC sampler converge if the overall sampler is still far from its target. The "classical" way of running interleaved MCMC samplers, such as MDA-P, is to stay for some time at each of the sub-MCMC samplers, and then "leave" the samplers, returning the last realisation as the result of the draw. The reason is that although convergence is not required every time, we don't want to baby-step through the state-space either. It is only reasonable to let both chains adapt a bit to the ever changing value on the conditional side, model or impu-

tation, by giving the chain the opportunity to move slightly towards a new region in the state space, before returning with the definite draw.

In our MDA-P context, the classical approach implies entering the I-step conditional on the output of the last MP-step, staying for some time at the I-step letting the chain adapt, and then return the last imputation as the result. The MP-step is now entered conditional on the imputation just returned from the I-step. We stay at the MP-step for while letting the chain adapt to the new input, and then return the last model as the result of the draw, and so on. Obviously, for MDA-P this classical approach is computationally expensive, because both sub-MCMC samplers are non-trivial MCMC samplers on their own. If we opt for the alternative, i.e., create only one realisation at every step, the overall sampler will mix extremely slowly, and there is hardly any progress. In practice this makes MDA-P useless because the sampler will get "lost in detail" and it is more prone to getting stuck in sub-optimal regions of the state space for long periods of time.

In the classical approach, the realisations generated between entering and leaving the sub-MCMC samplers are "wasted" in the sense that they are not used by the other sub-MCMC sampler. Those samples serve only as interim results for the final "return value". There is however an important observation with regard to those interim samples produced: the correlation between the realisations (imputations or models) means that there is only a gradual improvement in the samples as time passes; there is no clear threshold from which point in time samples may be considered as being "correct samples". The value returned by the sub-MCMC samplers is therefore to some degree rather arbitrary. In fact, there may be several of the interim draws that would be equally well suited as the return value. As such these samples have not been exploited to their full extent, which is unfortunate because we may have put a lot of computational effort into generating them. Moreover, many samples may be suitable across several iterations of the MDA-P sampler because MCMC works by step-wise exploration of the state space rather than taking sudden jumps. If we use the MCMC "walking around the state space" metaphor, we have that the chain within a certain time interval visits the same states several times, before gradually wandering off to some other region; there are always several samples that are suitable for a certain number of iterations.

Once we draw models in the MP-step, a lot of effort has been put into generating completions of the data in the I-step, before returning the final $\boldsymbol{U}^{(t)}$ that acts as input for $\Pr(M|\boldsymbol{U}^{(t)}, \boldsymbol{o})$. After returning this

last imputation as the final result of the I-step, the intermediate results are discarded, although they may be quite similar (in distribution that is) to the realisations that would have produced for the next call of the I-step given the next model. Any interim result may prove to be a reasonable "return value" that could serve as input for the next MP-step, especially because the models only differ slightly between the MP-steps due to correlation.

3.5 Imputation via importance sampling

The approach we propose in this section deals with the issue of "wasted imputations" by re-using imputations generated the I-phase. We collect and save the imputation samples we come across in a region that is currently being explored by the Markov chain, because these samples are all "good" samples for a certain amount of time (iterations). Obviously we can't save all samples we come across, so only a few imputations are saved. Using importance sampling we are able to use the same imputation for some time as the chain gradually moves away from the region to explore other parts of the space. The proposed method allows us to run MDA-P the classical way because we don't have to generate several completions after each model draw anymore, but re-use many imputations. This section is based on Riggelsen and Feelders, 2005.

3.5.1 The general idea

The general idea is based on the notion of a "population" of realisations, where each realisation is an imputation or completion of the incomplete data drawn from the predictive distribution. To a realisation, a weight is attached that determines its importance compared to the other realisations in the population. For model learning from incomplete data, the model changes dynamically, and therefore the weight of each realisation in the population is re-evaluated such that the population at any time reflects the correct predictive distribution. This re-weighing allows us to reuse the same realisations as new models are drawn.

Re-weighing is less expensive than producing new realisations all the time. Combined with the fact that imputations may be useful for several neighbouring models, it provides a way of alleviating the problem of "wasted imputations".

3.5.2 Importance sampling in the I-step—ISMDA-P

The I-step of MDA-P consist of drawing from $\Pr(\boldsymbol{U}|\boldsymbol{o}, M^{(t)}, \hat{\boldsymbol{\theta}}^{(t)})$, where $\hat{\boldsymbol{\theta}}^{(t)} = \mathrm{E}[\boldsymbol{\Theta}|\boldsymbol{o}, \boldsymbol{U}^{(t)}]$. Instead of drawing from $\Pr(\boldsymbol{U}|\boldsymbol{o}, M^{(t)}, \hat{\boldsymbol{\theta}}^{(t)})$, consider the sampling distribution $\Pr(\boldsymbol{U}|\boldsymbol{o}, M^{(t-q)}, \hat{\boldsymbol{\theta}}^{(t-q)}), q < t$, i.e., the "correct" distribution at time $t-q$ is used as the sampling distribution (the "wrong" distribution) at a later time t. We assume that the distributions at time t and time $t-q$ do not differ substantially—more on this in Section 3.5.3.

We create a "population" of empirical samples from the sampling distribution at time $t-q$, consisting of $\boldsymbol{u}^{(1)}, \ldots, \boldsymbol{u}^{(n)}$. At time t, we want to use these samples as if they were sampled from $\Pr(\boldsymbol{U}|\boldsymbol{o}, M^{(t)}, \hat{\boldsymbol{\theta}}^{(t)})$, and we therefore correct for the mismatch between realisations from $\Pr(\boldsymbol{U}|\boldsymbol{o}, M^{(t)}, \hat{\boldsymbol{\theta}}^{(t)})$ and $\Pr(\boldsymbol{U}|\boldsymbol{o}, M^{(t-q)}, \hat{\boldsymbol{\theta}}^{(t-q)})$ via the following importance weights:

$$w_i = \frac{\Pr(\boldsymbol{u}^{(i)}|\boldsymbol{o}, M^{(t)}, \hat{\boldsymbol{\theta}}^{(t)})}{\Pr(\boldsymbol{u}^{(i)}|\boldsymbol{o}, M^{(t-q)}, \hat{\boldsymbol{\theta}}^{(t-q)})}$$

The normalised importance weight for the ith draw becomes:

$$\frac{w_i}{W} = \frac{w_i}{\sum_{r=1}^{n} w_r} \tag{5.7}$$

By filling in, we may write eq. 5.7 as:

$$\begin{aligned}
\frac{w_i}{\sum_{r=1}^{n} w_r} &= \frac{\dfrac{\Pr(\boldsymbol{o}, \boldsymbol{u}^{(i)}|M^{(t)}, \hat{\boldsymbol{\theta}}^{(t)}) / \Pr(\boldsymbol{o}|M^{(t)}, \hat{\boldsymbol{\theta}}^{(t)})}{\Pr(\boldsymbol{o}, \boldsymbol{u}^{(i)}|M^{(t-q)}, \hat{\boldsymbol{\theta}}^{(t-q)}) / \Pr(\boldsymbol{o}|M^{(t-q)}, \hat{\boldsymbol{\theta}}^{(t-q)})}}{\displaystyle\sum_{r=1}^{n} \frac{\Pr(\boldsymbol{o}, \boldsymbol{u}^{(r)}|M^{(t)}, \hat{\boldsymbol{\theta}}^{(t)}) / \Pr(\boldsymbol{o}|M^{(t)}, \hat{\boldsymbol{\theta}}^{(t)})}{\Pr(\boldsymbol{o}, \boldsymbol{u}^{(r)}|M^{(t-q)}, \hat{\boldsymbol{\theta}}^{(t-q)}) / \Pr(\boldsymbol{o}|M^{(t-q)}, \hat{\boldsymbol{\theta}}^{(t-q)})}} \\
&= \frac{\Pr(\boldsymbol{o}, \boldsymbol{U}^{(i)}|M^{(t)}, \hat{\boldsymbol{\theta}}^{(t)}) / \Pr(\boldsymbol{o}, \boldsymbol{u}^{(i)}|M^{(t-q)}, \hat{\boldsymbol{\theta}}^{(t-q)})}{\sum_{r=1}^{n} \Pr(\boldsymbol{o}, \boldsymbol{u}^{(r)}|M^{(t)}, \hat{\boldsymbol{\theta}}^{(t)}) / \Pr(\boldsymbol{o}, \boldsymbol{u}^{(r)}|M^{(t-q)}, \hat{\boldsymbol{\theta}}^{(t-q)})}
\end{aligned}$$

It follows that we may reduce the importance weights to the likelihood ratio:

$$w_i = \frac{\Pr(\boldsymbol{o}, \boldsymbol{u}^{(i)}|M^{(t)}, \hat{\boldsymbol{\theta}}^{(t)})}{\Pr(\boldsymbol{o}, \boldsymbol{u}^{(i)}|M^{(t-q)}, \hat{\boldsymbol{\theta}}^{(t-q)})} \tag{5.8}$$

We have a closed form for the likelihood of complete data given in eq. 3.1, and therefore this ratio is easy to compute.

From Chapter 4, Section 1.1 it follows that we indeed approximate $\Pr(\boldsymbol{U}|\boldsymbol{o}, M^{(t)}, \hat{\boldsymbol{\theta}}^{(t)})$ for $n \to \infty$ because $\Pr(\boldsymbol{U}|\boldsymbol{o}, M^{(t-q)}, \hat{\boldsymbol{\theta}}^{(t-q)}) > 0$

whenever $\Pr(\boldsymbol{U}|\boldsymbol{o}, M^{(t)}, \hat{\boldsymbol{\theta}}^{(t)}) > 0$ since $\boldsymbol{\theta} > \mathbf{0}$, i.e., the parameter is always strictly positive.

The sampling process, which we refer to as ISMDA-P (Importance Sampling MDA-P), now proceeds as follows:

---

1 ISI-step: In the *importance sampling I-step* perform the following:

   (a) If the distributions at time $t-q$ and t differ substantially, then replace population by $\boldsymbol{u}^{(1)}, \ldots, \boldsymbol{u}^{(n)}$ via Gibbs sampling $\boldsymbol{U}^{(i)} \sim \Pr(\boldsymbol{U}|\boldsymbol{o}, m^{(t)}, \hat{\boldsymbol{\theta}}^{(t)})$, and compute $\rho_i = \Pr(\boldsymbol{o}, \boldsymbol{u}^{(i)}|m^{(t)}, \hat{\boldsymbol{\theta}}^{(t)})$. Otherwise reuse population (and likelihoods) from last ISI-step.

   (b) For all $\boldsymbol{u}^{(i)}$ compute the likelihood $\tau_i = \Pr(\boldsymbol{o}, \boldsymbol{u}^{(i)}|M^{(t)}, \hat{\boldsymbol{\theta}}^{(t)})$.

   (c) Compute the importance weights $w_i = \tau_i/\rho_i$, and $W = \sum_{r=1}^{n} w_r$.

   (d) With probability w_i/W select $\boldsymbol{u}^{(i)}$, i.e., select a completion according to the normalised importance weights.

2 MP-step: In the *model posterior* step draw $M^{(t+1)} \sim \Pr(M|\boldsymbol{o}, \boldsymbol{u}^{(i)})$ via MCMC, by providing the model $m^{(t)}$ as the seed. Also compute $\hat{\boldsymbol{\theta}}^{(t+1)} = \mathrm{E}[\boldsymbol{\Theta}|\boldsymbol{o}, \boldsymbol{u}^{(i)}]$.

THEOREM 5.3 ISMDA-P *produces an ergodic Markov chain with invariant distribution* $\Pr(M|\boldsymbol{o})$ *when in the ISI-step* $n \to \infty$.

For the MP-step ISMDA-P is the same as for MDA-P. To test for irreducibility in the ISI-step, we need to makes sure that all of $\Omega_{\boldsymbol{U}}$ can be reached. Since both the numerator $\Pr(\boldsymbol{U}|\boldsymbol{o}, M^{(t-q)}, \hat{\boldsymbol{\theta}}^{(t)}) > 0$ and the denominator $\Pr(\boldsymbol{U}|\boldsymbol{o}, M^{(t-q)}, \hat{\boldsymbol{\theta}}^{(t)}) > 0$, because $\boldsymbol{\theta} > \mathbf{0}$, we have that for $n \to \infty$, the entire finite state space is reached. Aperiodicity is guaranteed because there is a non-zero probability of sampling the same value again.

We emphasise that this result is quite uninteresting from a practical point of view, since n is usually kept relatively small. Intuitively speaking, we should just make sure that the n samples at any time represents the distribution in "area" we are exploring with the sampler. More on this in the next section.

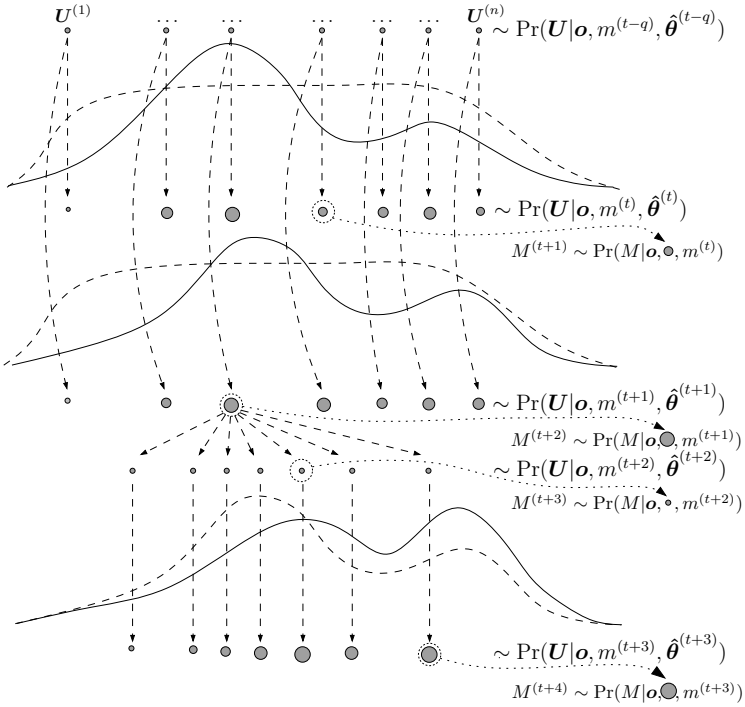

Figure 5.2. Population based re-weighing scheme. By way of importance sampling the imputations are re-used for different DAG models as time passes. A new population of imputations is generated after some time.

Figure 5.2 illustrates the process. The dashed line in the top plot depicts the sampling distribution at time $t - q$, and the solid line the correct distribution at time t, i.e., the distribution from for which we require samples. Initially n draws are made from $\Pr(\boldsymbol{U}|\boldsymbol{o}, m^{(t-q)}, \hat{\boldsymbol{\theta}}^{(t-q)})$, and they are assigned an importance weight (depicted as a solid circle indicating mass). Depending on the weight, $\boldsymbol{u}^{(i)}$ is selected, and the complete data $(\boldsymbol{o}, \boldsymbol{u}^{(i)})$ is used for drawing a model at time $t + 1$. This model gives rise to the middle plot, which is the new distribution from which we require samples. We reuse the samples from time $t - q$ from the first plot, and only compute the new importance weights, draw a completion, and draw a model at time $t + 2$. After having drawn that model, a new population is sampled from $\Pr(\boldsymbol{U}|\boldsymbol{o}, m^{(t+2)}, \hat{\boldsymbol{\theta}}^{(t+2)})$, and the process continues: a model is drawn at time $t + 3$ giving rise to the distribution $\Pr(\boldsymbol{U}|\boldsymbol{o}, m^{(t+3)}, \hat{\boldsymbol{\theta}}^{(t+3)})$. This distribution is approximated by using the population that was sampled at time $t-2$. A completion is

selected depending on the importance weights, and finally at time $t+4$ a model is drawn again.

3.5.3 Generating new population vs. re-weighing

Since ISMDA-P employs importance sampling, the remarks made in Chapter 4, Section 1.1.1 apply here as well. The population together with the importance weights is the empirical evidence we have of the distribution from which we require realisations. Depending on the importance weights, the population may be an extremely crude approximation. As discussed in Chapter 4, Section 1.1.1, the similarity between the sampling distribution and the distribution from which we want empirical samples is crucial for a decent approximation. This is exactly why we require that the distributions at time $t-q$ and $t, t>q$ should be "close". For large q, the difference may be substantial, in which case it is insufficient to merely re-weigh the population, as none of its members may lie in a region of mass, but in some tail region. In that case a new population should be generated, after which it for some time again suffices to re-weigh the population.

The question is then how we should monitor the population and diagnose a "dying" one. The following alternatives seem plausible:

- Define a similarity measure between models $\delta(M_1, M_2)$ in terms of the edges of the DAG models. If the difference between M_1 and M_2 is above a certain threshold then the number of (in)dependences may be too large, and the distributions may be very dissimilar.
- Monitor the variance of the normalised importance weights. If the variance becomes too large, it indicates that only few weights "carry the burden".
- Generate a new population for every qth iteration, i.e., cycle between the ISI-step and the MP-step q times, and then create a new population. Due to the nature of MCMC samplers, it may be a good idea to change q dynamically. For instance, in the burn-in phase the population should be renewed more often.

We adopt the last option, because it is easy to implement in practice. This simple approach in general means that the sampler has to be run a couple of times such that q can be chosen empirically.

It should be stressed that even when a population does not seem to be dying according to any diagnostic, there is still a need to renew the population once in while. Eventually the completions have been

exploited exhaustively, and in order to reach the entire space of possible completions, the space should also be *explored*, and new realisations should be generated. If the samples are not renewed we risk that the ISMDA-P sampler becomes reducible, because certain regions of the state space are never considered.

3.5.4 The marginal likelihood as predictive distribution

As we saw in Chapter 3, Section 3.2.1, the marginal likelihood plays a central role in model learning within the Bayesian approach. For model learning the intended reading of $\Pr(\boldsymbol{o}, \boldsymbol{U}|M)$ is "the marginal likelihood of M given $\boldsymbol{o}, \boldsymbol{U}$". However, as it stands, the marginal likelihood is precisely the probability of $(\boldsymbol{o}, \boldsymbol{U})$ given model M. This means that *given* M, we can measure how well a given data sample adheres to the (in)dependences that M encodes. This measure is a probability that depends solely on the (in)dependence assumptions between the variables in $\boldsymbol{X}$, but is independent of the "strengths" that these (in)dependences have, i.e., $\boldsymbol{\Theta}$ is irrelevant.

Hence, if we have a method that produces data sets, we can measure how probable this sample is given M. In particular, because $\boldsymbol{o}$ is given already, we need only fill in $\boldsymbol{U}$ to produce a full data sample, and the overall probability of the augmented sample is $\Pr(\boldsymbol{o}, \boldsymbol{u}|M)$. The probability of $\boldsymbol{u}$ only given M is obtained by normalisation over possible assignments to $\boldsymbol{U}$.

The above discussion gives rise to an alternative to the ISMDA-P sampling scheme presented in Section 3.5.2. Up until now the distribution in the ISI-step was approximated with importance sampling using a sampling distribution from an earlier iteration. The approximation is of $\Pr(\boldsymbol{U}|\boldsymbol{o}, M^{(t)}, \hat{\boldsymbol{\theta}}^{(t)})$, i.e., a distribution conditional on $\boldsymbol{\Theta}$. Via the importance weights given in eq. 5.8, we need not compute the actual probability of the imputations, i.e., $\Pr(\boldsymbol{u}^{(i)}|\boldsymbol{o}, M^{(t)}, \hat{\boldsymbol{\theta}}^{(t)})$ but it suffices to compute $\Pr(\boldsymbol{u}^{(i)}, \boldsymbol{o}|M^{(t)}, \hat{\boldsymbol{\theta}}^{(t)})$. This probability can be interpreted as a measure of *imputation quality* given the current $\hat{\boldsymbol{\theta}}^{(t)}$. However, we actually have a closed expression for imputation quality unconditional on $\boldsymbol{\Theta}$, namely the marginal likelihood $\Pr(\boldsymbol{U}^{(i)}, \boldsymbol{o}|M^{(t)})$. By plugging in $\Pr(\boldsymbol{u}^{(i)}, \boldsymbol{o}|M^{(t)})$ as the numerator in eq. 5.8, we approximate $\Pr(\boldsymbol{U}|\boldsymbol{o}, M^{(t)})$ at every ISI-step:

$$w_i = \frac{\Pr(\boldsymbol{o}, \boldsymbol{u}^{(i)}|M^{(t)})}{\Pr(\boldsymbol{o}, \boldsymbol{u}^{(i)}|M^{(t-q)}, \hat{\boldsymbol{\theta}}^{(t-q)})}$$

This means that we effectively skip many intermediate approximations of $\Pr(\boldsymbol{U}|\boldsymbol{o}, M^{(t)}, \hat{\boldsymbol{\theta}}^{(t)})$ "on our way to" to $\Pr(\boldsymbol{U}|\boldsymbol{o}, M^{(t)})$. Instead we approximate $\Pr(\boldsymbol{U}|\boldsymbol{o}, M^{(t)})$ directly at every iteration. In other words, we are now actually applying the following Gibbs sampler:

$$\begin{aligned} \boldsymbol{U}^{(t+1)} &\sim \Pr(\boldsymbol{U}|\boldsymbol{o}, m^{(t)}) \\ M^{(t+1)} &\sim \Pr(M|\boldsymbol{o}, \boldsymbol{u}^{(t+1)}) \\ &\vdots \end{aligned}$$

where $\Pr(\boldsymbol{U}|\boldsymbol{o}, M^{(t)})$ is approximated via importance sampling.

THEOREM 5.4 ISMDA-P *with the marginal likelihood in the ISI-step produces an ergodic Markov chain with invariant distribution* $\Pr(M|\boldsymbol{o})$ *when in the ISI-step* $n \to \infty$.

Since $\Pr(\boldsymbol{U}, \boldsymbol{o}|M) > 0$ we may use the same reasoning as we did for ISMDA-P. However, we stress again that from a practical point of view the significance of this theorem is less interesting because n is kept relatively small in practice.

3.5.5 The *e*MC^4 sampler

Following the tradition of the MCMC model sampling algorithms for complete data, called (e)MC^3, we refer to marginal likelihood version of ISMDA-P as *(enhanced) Markov Chain Monte Carlo Model Composition with Missing Components*, (e)MC^4.

Algorithm 2 contains the pseudocode of the sampler. Lines 5–9 create a new population when required, by drawing from the current model and the current expected parameter value. Also the likelihood is computed and saved. In lines 10–14 the completions are evaluated via the marginal likelihood using the current model, and the importance weights are computed and normalised. In line 15 a completion is selected from the population depending on its importance weight. The selected completion is used in conjunction with the observed part of the data to call an MCMC model sampler in line 16, where a new model is drawn. The initial seed for the MCMC model sampler is the last model at time t. Finally in line 17, the model is saved as a realisation from $\Pr(M|\boldsymbol{O})$.

There is one important point about line 12 that we would like to emphasise. At first sight it may seem rather expensive to even compute τ_i for each imputation $\boldsymbol{U}^{(i)}$ at every iteration. However, since $M^{(t-1)}$ and $M^{(t)}$ are correlated, they differ in only a few adjacencies. This means

Algorithm 2: eMC^4

Input : n, population size (no. of completions kept).
Onput: DAG models from $\Pr(M|\boldsymbol{o})$ (requires burn-in).

$m^{(0)} \leftarrow G = (\boldsymbol{X} = \{X_1, \ldots, X_p\}, \boldsymbol{E} = \varnothing)$
$\boldsymbol{u}^{(0)} \leftarrow$ valid random completion
$r \leftarrow 0$
for $t \leftarrow 0$ **to** ∞ **do**
  /* MCMC draw imputations (population) */
  if *population-is-dead* **or** *exhaustively-exploited* **then**
    $\hat{\boldsymbol{\theta}} = \mathrm{E}[\boldsymbol{\Theta}|\boldsymbol{o}, \boldsymbol{u}^{(r)}]$
    for $i \leftarrow 1$ **to** n **do**
      draw $\boldsymbol{U}^{(i)} \sim \Pr(\boldsymbol{U}|\boldsymbol{o}, m^{(t)}, \hat{\boldsymbol{\theta}})$
      $\rho_i \leftarrow \Pr(\boldsymbol{u}^{(i)}, \boldsymbol{o}|m^{(t)}, \hat{\boldsymbol{\theta}})$
  /* Compute importance weights */
  $W \leftarrow 0$
  for $i \leftarrow 1$ **to** n **do**
    $\tau_i \leftarrow \Pr(\boldsymbol{o}, \boldsymbol{u}^{(i)}|m^{(t)})$
    $w_i \leftarrow \tau_i/\rho_i$
    $W \leftarrow W + w_i$
  draw $R \sim \Pr(i) \leftarrow w_i/W$
  /* Few MCMC iterations, return last as $M^{(t+1)}$ */
  draw $M^{(t+1)} \sim \Pr(M|\boldsymbol{o}, \boldsymbol{u}^{(r)}; m^{(t)})$ /* seed with $m^{(t)}$ */
  record($m^{(t+1)}$)

that most vertices have the same parent sets. Since τ_i (the marginal likelihood) decomposes, we only need to re-compute the imputation quality score for the vertices that have different parent sets. Hence, for all $i = 1, \ldots, n$, $\tau_i = \prod_{j=1}^{p} \tau_i(X_j)$, save between iterations the imputation quality score $\tau_i(X_j)$ for each vertex. Only when the entire population is renewed τ_i needs to be computed for every vertex.

3.5.6 Evaluation—proof of concept

In this section we perform a small experimental evaluation of eMC^4 and briefly discuss the results.

We used a data set from Edwards and Havránek, 1985 about probable risk factors of coronary heart disease. The data set consists of 1841

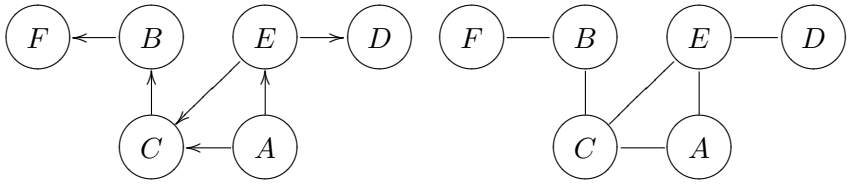

Figure 5.3. Generating model. *Left:* DAG model. *Right:* essential graph model.

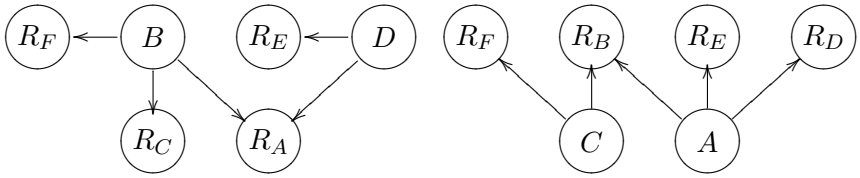

Figure 5.4. Missing data mechanisms. *Left:* mechanism 1. *Right:* mechanism 2.

records and 6 binary variables, A: smoking, B: strenuous mental work, C: strenuous physical work, D: blood pressure under 140, E: ratio β to α proteins less than 3, F: family history of coronary heart disease. Because several DAGs encode the same set of assumptions about independence, we depict results as essential graphs.

Based on an eMC^3 run using the 1841 complete records, the model on the left in figure 5.3 is a highly probable model (although edge $F - B$ is not strongly supported). The model on the right is the corresponding essential graph of the DAG. The parameter corresponding to the DAG model was determined based on the aforementioned data set, and 1800 new records were sampled from the BN.

Incomplete sets were generated by applying missingness mechanism one in figure 5.4 on the complete sample. This graph explicitly defines how response R_i of variable i depends on observed variables. Since for all X_i, R_i only depends on completely observed variables, the missingness mechanism is clearly ignorable. Three incomplete sets were generated with 5–10%, 10–15% and 15–20% missing components. The probability of non-response of variable X_i conditional on a parent configuration of R_i was selected from the specified interval.

On the basis of the generating model and the missingness mechanism, we would expect the following results. Since response of C only depends on B and the association $C - B$ is strong, a big fraction of components can be deleted for C without destroying support for the edge in the data. Association $D - E$ is also strong so discarding components for E

will probably not have a major impact on the edge either. Association $E - A$ is influenced by B and D because the response is determined by those variables. Values for E and A may be absent often and therefore information about the association might have changed. This may also be the case for the edges $C - E$ and $C - A$.

We ran eMC^4 using each incomplete data set. For drawing a model from the model posterior, 150 model Metropolis-Hastings steps were performed, and the last model was returned as the final result of the draw. The size of the population was set to 25, and it was refreshed, i.e., old population discarded and new one drawn, after every 40th model drawn, i.e., $q = 40$.

In figure 5.5 the top four models are depicted along with their sampling frequencies. Notice the presence of the strong associations $C - B$ and $D - E$ everywhere, as expected. When the fraction of missing components for two associated variables increases it has a big impact on the support of such an association. Indeed, from the figure we see that the support for associations between variables A, C and E has changed. The sample frequencies and the number of visited models also suggest that the variance of the posterior distribution becomes bigger when more components are deleted. There is no longer a pronounced 'best' model.

The plot in figure 5.6 shows this more clearly. Here the cumulative frequencies are plotted against models (sorted on frequency in descending order). A steep plot indicates a small variance. For complete data the 10 best models account for 90% of the distribution whereas for 15–20% missing components only 50% of the distribution is accounted for by the best 10 models.

To investigate the similarity of the models between the three incomplete sets, we used eq. 4.12, where $\Delta(M)$ is set to 1 when there is an edge between two vertices of interest in M. This results in the expected probability of the presence of edges as seen in figure 5.6, bottom. We can see, as we would expect, that the distance between points of complete data and incomplete data is dependent on the fraction of missing components. Diamonds (15–20%) have the biggest distance to triangles (complete), and plusses (5–10%) the smallest.

As we saw for mechanism one, discarding components for two associated variables can have a big impact on the presence of the corresponding edge in sampled models. For strongly associated variables the impact is less pronounced. We created another incomplete data set using mechanism two in figure 5.4. For the associated variables C, E and A the mechanism only discards components that we think will not severely

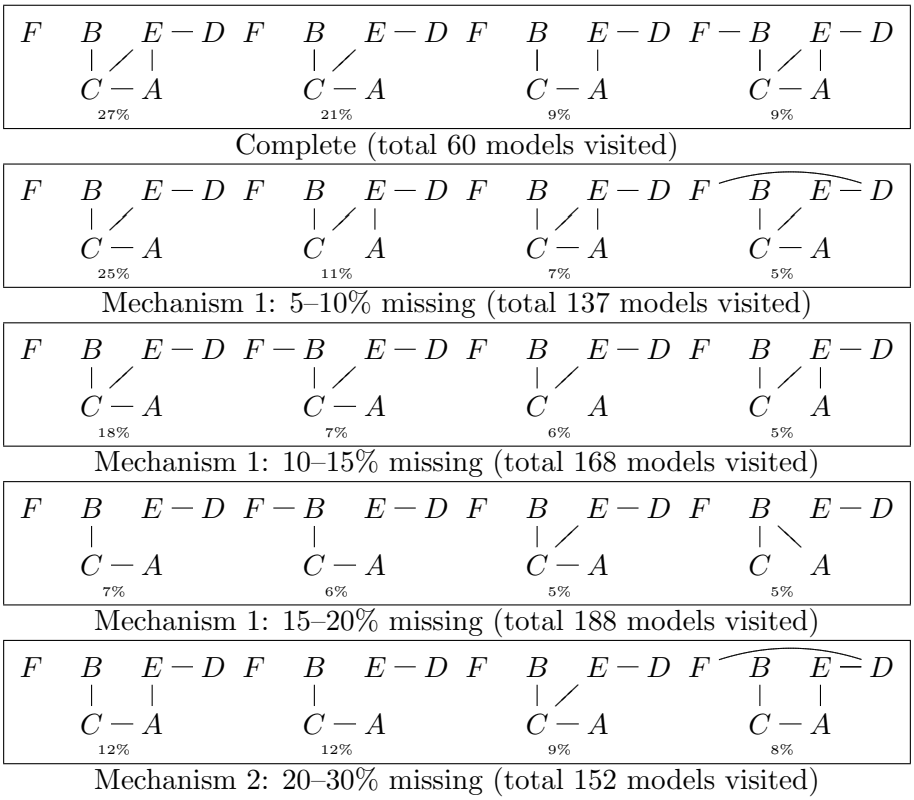

Figure 5.5. Top four visited models and sampling frequencies. Note that all edges are reversible.

impact these associations. For the strong association $E - D$ discarding components on both should not matter. We expect that we are able to remove a substantial fraction of components and still obtain reasonable models. We selected the fraction of missing components in the interval 20–30%. In the last row in figure 5.5 we see that although a substantial fraction of components were deleted, the models learned are quite similar to the models from the complete set.

To illustrate that it is not the fraction of missing components that determines the variance but rather the fraction of missing information (Little and Rubin, 1987), we plotted the cumulative frequency in figure 5.6, top. The variance of the posterior distribution is similar to the variance of the posterior for mechanism one with 5–10% missing components. This means that although the fraction of missing components

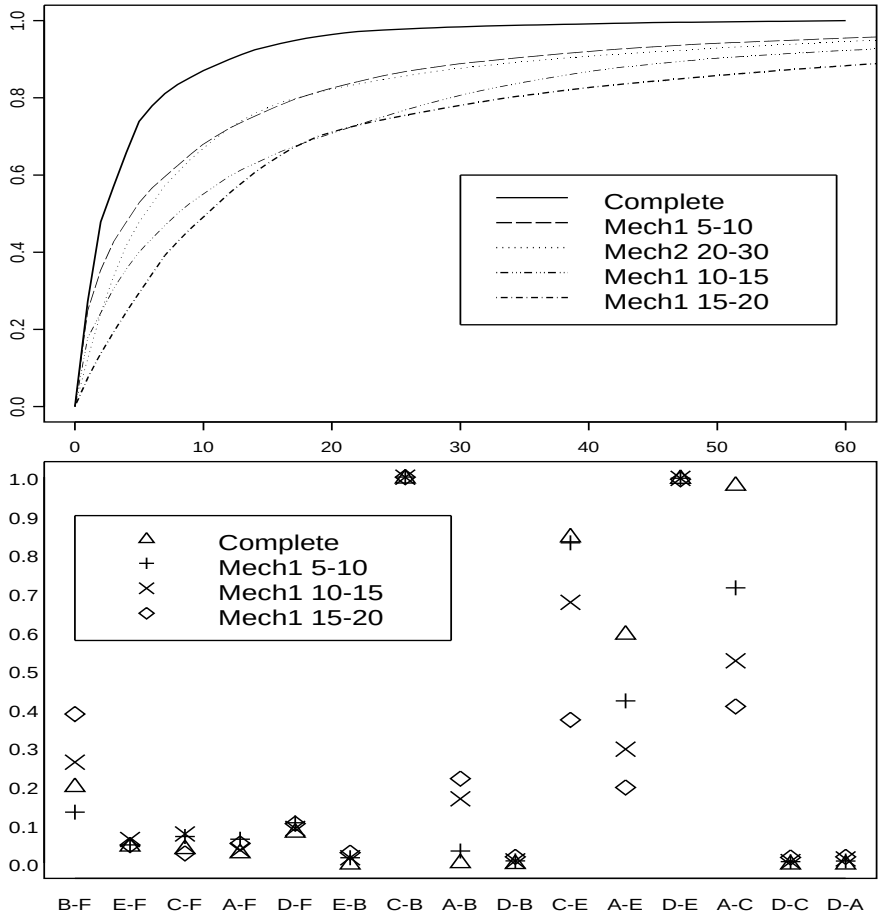

Figure 5.6. *Top*: cumulative frequencies. *Bottom*: expected probability of edges.

is much higher than 5–10%, the uncertainty due to missing data has not changed substantially.

Finally we compare $e\text{MC}^4$ in a model selection setting. The four most probable models in figure 5.5 would be the models selected as the MAP of $\Pr(M|\boldsymbol{o})$. For the data sets with 5–10%, 10–15% and 15–20% missing components, we ran the SEM implementation by Friedman and Elidan, 2005. The SEM-algorithm was run with default parameters, except for the ESS which we set to 1 (the default is 5). In figure 5.7 the models selected are shown.

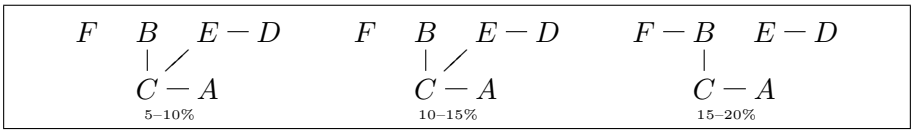

Figure 5.7. The models selected with the SEM-algorithm.

For the two data sets with 5–10% and 10–15% missing components, the models selected coincide with the two most frequently sampled models using eMC^4. For 15–20% missing components, SEM selects the 2nd most frequently sampled model. We see that the top 4 most probable models are visited almost equally often using eMC^4, and SEM selects an arbitrary one. This nicely illustrates the added benefit of taking a Bayesian approach, since we now get an impression of the structural differences between models that score almost equally well.

3.5.7 Conclusion

To our knowledge eMC^4 is the first and only Bayesian approach to learning Bayesian network models from incomplete data. The reason for this we believe is that a Bayesian approach via "straightforward" MCMC simulation is computationally extremely expensive. An even larger obstacle is that it is in practice almost impossible to get such MCMC samplers to converge and mix properly. Even with a more clever MCMC approach such as eMC^4, it is still relatively expensive to sample models from the posterior model distribution, but the mixing is improved.

Valuable insight is gained when sampling models from the posterior model distribution; an illustration of the kind of information one can derive from posterior realisations is given in the previous section. A posterior distribution is more informative than just a single model. This is especially true in the case of incomplete data, since the increased uncertainty due to missing data is reflected in the probability distribution. The question is then if the insight a Bayesian statistical approach offers is "worth the effort" from computational point of view.

In the context of parameter learning of BNs, a related approach to eMC^4 can be employed, but then for a fixed DAG model. This means that a broad range of approximate predictive distributions can be used, and the "error" that the approximation introduces is corrected for by importance sampling in the ISI-step. We refer to Riggelsen, 2006b for details on this.

4. Ad-hoc and heuristic methods

There are several techniques for handling the problem of missing data that are less principled in nature; see for instance Allison, 2001. The methods are often ad hoc, and for several techniques it is not evident what is actually gained or lost by applying them without a thorough analysis of the assumptions underlying the technique. Sometimes ad-hoc methods may provide a good enough approximation of what would have been obtained by a principled analysis. A major problem with many ad-hoc techniques is that they fail to preserve the dependence relationship between the variables. It is not enough to focus only on the "column" for which data items are missing in order to predict the unobserved items. This in general is the easiest thing to do, but the multivariate relationship among variables is then distorted and severe bias may be the result.

There may be several reasons as to why heuristics or ad-hoc approaches are used. The methods and algorithms discussed so far are iterative in nature, they require several passes through the data. At every iteration, an ever "better" predictive distribution is used for obtaining the sufficient statistics corresponding to completions of the incomplete data. Generating these completions either requires exact inference in a BN or application of a Gibbs sampler, which generally speaking is considered computationally demanding. Most heuristic and ad-hoc approaches require far less passes, because they are generally not based on the notion of "refining" a predictive distribution such that the predictions become more and more "correct".

If for some reason it is undesirable to traverse the data too often, we probably also have to sacrifice a Bayesian approach to data analysis, because a Bayesian approach normally implies *many* traversals. This is not as bad as it seems because for large amounts of data, posteriors will approximate Normal distributions (Carlin and Louis, 2000) anyway. In that case we may settle for summary statistics, such as the MAP or the expectation. This means that model selection becomes interesting again. Most ad-hoc approaches exactly produce summary statistics, albeit approximate ones.

4.1 Available cases analysis

In Section 1.1 we briefly mentioned the very simple approach referred to as *complete case analysis*, which simply deletes the cases in $\boldsymbol{d}$ that contain missing values. This approach is only valid, albeit inefficient,

if the missing data mechanism is MCAR (Allison, 2001; Brand, 1999). A slight variation on this theme called *available cases analysis* takes advantage of the fact that different parameters require different data items in $\boldsymbol{d}$. For estimating $\theta_{X_i|\boldsymbol{X}_{pa(i)}}$ only variable X_i and the variables in the parent set $\boldsymbol{X}_{pa(i)}$ are needed. The estimation is then performed by considering cases for which these variables are observed simultaneously, and the remaining cases are discarded (hence also assumes MCAR). This is done for all parameters of the BN. The advantage of this approach is that less information is discarded than when entire records are neglected.

Once the parameter has been estimated as $\hat{\boldsymbol{\theta}}$, the predictive distribution becomes $\Pr(\boldsymbol{U}|\boldsymbol{o}, m, \hat{\boldsymbol{\theta}})$. Using this distribution, the incomplete data can be completed. Alternatively, if it is only the parameter of the BN that is required, we can use $\hat{\boldsymbol{\theta}}$ directly as a rough approximation of $\boldsymbol{\theta}$.

4.2 Bound and Collapse—BC

The Bound and Collapse algorithm (BC) (Ramoni and Sebastiani, 1997a; Ramoni and Sebastiani, 1997b) is a method claimed to be able to return not necessarily the ML estimate but still qualified guesses as to what the parameter of a BN could be. In that sense BC is not really a valid statistical method of learning the BN parameter. The algorithm is a deterministic non-iterative two-phase process: The *bound* phase calculates the extreme bounds for the sufficient statistics consistent with the data sample. On the basis on these statistics, bound parameters are easily calculated (they are simply a function of the bound statistics). The *collapse* phase is a convex combination of these extreme parameters (for a given child/parents configuration) where the coefficients for the bounds are based on an available cases analysis.

The sufficient statistics $n(x_i, \boldsymbol{x}_{pa(i)})$ is the number of occurrences of the configuration $x_i, \boldsymbol{x}_{pa(i)}$ in the data. When dealing with incomplete data, this function is not defined for cases with unknow entries either because X_i is missing, $\boldsymbol{X}_{pa(i)}$ is (perhaps partly) missing or a combination of the two.

We now re-defined $n(x_i, \boldsymbol{x}_{pa(i)})$ to pertain to that part of the incomplete sample, where $X_i, \boldsymbol{X}_{pa(i)}$ is *fully observed*. For the other part, where $X_i, \boldsymbol{X}_{pa(i)}$ is *not fully observed*, we define $\tilde{n}(x_i, \boldsymbol{x}_{pa(i)})$. This function counts the records that can be completed such that they match the configuration $x_i, \boldsymbol{x}_{pa(i)}$. Hence, $\tilde{n}(x_i, \boldsymbol{x}_{pa(i)})$ returns the number of cases that are consistent with $x_i, \boldsymbol{x}_{pa(i)}$, neglecting the cases for which

$X_i, \boldsymbol{X}_{pa(i)}$ are fully observed. We define $\tilde{n}(\boldsymbol{x}_{pa(i)}|x_i)$ in a similar fashion, but the bar means that we fix X_i at x_i; i.e., consider those records with incomplete observation for $\boldsymbol{X}_{pa(i)}$ that are consistent with $\boldsymbol{x}_{pa(i)}$ *given* that X_i is observed and has value x_i.

In order to find the ML estimates for the multinomial distribution we need only the sufficient statistics. Thus we can find the upper bound ML estimate for $\theta_{x_i|\boldsymbol{x}_{pa(i)}}$, called $\theta^{\bullet}_{x_i|\boldsymbol{x}_{pa(i)}}$, by using the statistics $\tilde{n}(x_i, \boldsymbol{x}_{pa(i)}) + n(x_i, \boldsymbol{x}_{pa(i)})$. This estimate is simply:

$$\theta^{\bullet}_{x_i|\boldsymbol{x}_{pa(i)}} = \frac{n(x_i, \boldsymbol{x}_{pa(i)}) + \tilde{n}(x_i, \boldsymbol{x}_{pa(i)})}{n(\boldsymbol{x}_{pa(i)}) + \tilde{n}(x_i, \boldsymbol{x}_{pa(i)})} \tag{5.9}$$

which gives us the parameter when all incomplete cases consistent with $x_i, \boldsymbol{x}_{pa(i)}$ are indeed assigned this configuration.

In Ramoni and Sebastiani, 2001 the lower bound ML estimate called $\theta_{\bullet x_i|\boldsymbol{x}_{pa(i)}}$ is defined such that the number of completions ascribed to $\boldsymbol{x}_{pa(i)}$ is maximal, without increasing the frequency of the configuration $x_i, \boldsymbol{x}_{pa(i)}$, i.e., for all records complete $\boldsymbol{x}_{pa(i)}$ except for records where X_i is observed as x_i:

$$\theta_{\bullet x_i|\boldsymbol{x}_{pa(i)}} = \frac{n(x_i, \boldsymbol{x}_{pa(i)})}{n(\boldsymbol{x}_{pa(i)}) + \tilde{n}(\boldsymbol{x}_{pa(i)}) - \tilde{n}(\boldsymbol{x}_{pa(i)}|x_i)} \tag{5.10}$$

The actual estimate $\hat{\theta}_{x_i|\boldsymbol{x}_{pa(i)}}$ lies somewhere between the upper bound $\theta^{\bullet}_{x_i|\boldsymbol{x}_{pa(i)}}$ and the lower bound $\theta_{\bullet x_i|\boldsymbol{x}_{pa(i)}}$. We collapse the bounds in eq. 5.9 and eq. 5.10 into one single estimate:

$$\hat{\theta}_{x_i|\boldsymbol{x}_{pa(i)}} = (1 - v(x_i)) \cdot \theta_{\bullet x_i|\boldsymbol{x}_{pa(i)}} + v(x_i) \cdot \theta^{\bullet}_{x_i|\boldsymbol{x}_{pa(i)}}$$

The first term corresponds to any completion of X_i other than x_i, and the second term to the completion x_i. The weight $v(\cdot)$ determines the exact location between the bounds (see bellow).

In Ramoni and Sebastiani, 1998 several lower bounds are defined for one upper bound. The lower bounds occur when the number of completions ascribed to $y_i, \boldsymbol{x}_{pa(i)}, y_i \neq x_i$ is maximal, i.e., for each completion y_i of X_i other than x_i:

$$\theta_{y_i \bullet x_i|\boldsymbol{x}_{pa(i)}} = \frac{n(x_i, \boldsymbol{x}_{pa(i)})}{n(\boldsymbol{x}_{pa(i)}) + \tilde{n}(y_i, \boldsymbol{x}_{pa(i)})} \tag{5.11}$$

The collapse is defined as a convex combination of all the lower bounds and the upper bound:

$$\hat{\theta}_{x_i|\boldsymbol{x}_{pa(i)}} = \sum_{y_i \neq x_i} v(y_i, \boldsymbol{x}_{pa(i)}) \cdot \theta_{y_i \bullet x_i|\boldsymbol{x}_{pa(i)}} + v(x_i, \boldsymbol{x}_{pa(i)}) \cdot \theta^{\bullet}_{x_i|\boldsymbol{x}_{pa(i)}}$$

The observed part of the data is considered a "representative" sample of the entire data. The weights $v(\cdot)$ is the relative (conditional) frequency of some configuration $v(x_i, \boldsymbol{x}_{pa(i)}) = n(x_i, \boldsymbol{x}_{pa(i)}) / \sum_{y_i} n(y_i, \boldsymbol{x}_{pa(i)})$ only considering cases where $x_i, \boldsymbol{x}_{pa(i)}$ is fully observed. For the unconditional version, $\boldsymbol{X}_{pa(i)}$ is empty.

If we want to use BC in a Bayesian setting, we have to account for prior counts $\boldsymbol{\alpha}$. This is done by adding the corresponding prior counts to the numerator and denominator of eq. 5.9, eq. 5.10 and eq. 5.11. This results in the expected parameter bounds rather than the ML parameter bounds.

In summary, BC consists of the following two steps:

---

1 B-phase: In the *bound* phase, the bounds of the parameter estimates are computed:

   (a) The upper bound is the maximum number of records consistent with $x_i, \boldsymbol{x}_{pa(i)}$.

   (b) The lower bound(s) depend(s) on the number of records that are consistent with $y_i, \boldsymbol{x}_{pa(i)}, y_i \neq x_i$. Either each completion with $y_i \neq x_i$ can be regarded as a lower bound, or a single lower bound can be defined, where the maximum number of records that can be ascribed to $\boldsymbol{x}_{pa(i)}$ is counted excluding those records for which X_i is observed with value x_i.

2 C-phase: In the *collapse* phase, determine the precise location of the parameter estimate between the lower and upper parameter estimate bounds. The collapse is based on the records for which $v(x_i, \boldsymbol{x}_{pa(i)})$ is fully observed.

BC is not an iterative procedure, and differs radically from EM and the MCMC methods previously discussed. The estimates found are not ML estimates, but are based on an ad hoc approach that from an intuitive point of view is attractive.

Essentially BC is a sophisticated available cases analysis, where attention is restricted to include only the child/parents pertaining to a particular parameter. Like available cases analysis, there is no predictive element involved in the process of obtaining the parameters with BC.

The interval obtained in the B-phase of BC may on its own be quite valuable. There is however no distribution or function defined on that interval that indicates the most probable area(s) within this interval that could form the basis of a parameter point estimate. The collapse phase is based on the frequency of X_i (and for the conditional collapse also on $\boldsymbol{X}_{pa(i)}$), estimated by taking only those records of the data into consideration where the variables are observed. When the weights in the collapse phase are based on only the marginal distribution of the child variable, i.e., by $v(x_i)$, it is questionable if this provides enough information to collapse the intervals pertaining to all the parameters $\theta_{x_i|\boldsymbol{X}_{pa(i)}}$ irrespective of the configuration assigned to $\boldsymbol{X}_{pa(i)}$. Moreover, in determining $v(\cdot)$ the underlying assumption is that the data is missing according to the MCAR assumption. When this does not hold, the collapse phase may become unreliable.

BC seems works for some missing data mechanisms, but unfortunately due to the points mentioned above, it can produce unpredictable results for others. Suppose that we want to estimate $\theta_{x_i|\boldsymbol{x}_{pa(i)}}$ from incomplete data, under the quite realistic MAR missing data mechanism where R_i depends on $\boldsymbol{X}_{ch(i)}$. BC may perform bad in this situation. The reason is that BC neglects any information beyond X_i and $\boldsymbol{X}_{pa(i)}$, which in this situation is required to determine $\hat{\theta}_{x_i|\boldsymbol{x}_{pa(i)}}$.

A desirable property of any incomplete data method, is that for equivalent DAG models, the same results should be expected. For available cases analysis and BC this does not hold.

4.3 Markov Blanket Predictor—MBP

Like BC, the method we describe in this section is a non-iterative procedure. To some degree the method resembles BC, but on crucial points it differs; most notably, we approach the missing data problem from a prediction point of view. Moreover, for equivalent models, the same results are obtained. This section is based on Riggelsen, 2006a.

4.3.1 The general idea

The basic idea behind our method, is to predict missing values on a per variable basis, by focusing only on those observed variables that directly influence the variables that are missing. The BN model explicitly identifies the relevant predictor variables as those variables that form the Markov blanket of a given variable. By looking for cases in the data that have similar values on the observed predictor variables, a predictive distribution for the missing data is created. When the number of predictors is too large, a selection is made, and only the best predictors (those that have the strongest influence) are considered when generating the predictive distribution.

4.3.2 Approximate predictive distributions

In this section we propose an approximation of the predictive distribution. The way the approximation is defined, leads to a non-iterative algorithm in a natural way. Additionally, there is no need to perform exact inference as in EM, or draw several realisations as in DA and its derivatives.

We assume that for any record l the following factorisation holds:

$$\Pr(\boldsymbol{U}^l|\boldsymbol{o}^l, M) = \prod_{k=1}^{r(l)} \Pr(U_k^l|\boldsymbol{o}^l, m) \tag{5.12}$$

That is, we are able to predict single missing values independently of each other through separate predictive distributions. If the independence assumptions expressed in eq. 5.12 actually do hold, then the dependence of U_k^l on $\boldsymbol{o}^l$ is in fact only on a subset of the observed variables, namely the Markov blanket of U_k^l.

If $X_i^l = U_k^l$, we can thus predict it if all variables in the Markov blanket are observed:

$$\Pr(X_i^l|\boldsymbol{o}^l, m) = \Pr(X_i^l|\boldsymbol{x}_{mb(i)}^l, m)$$

If not all variables of the Markov blanket are observed, then there are other observed variables outside the blanket that influence the prediction; this was illustrated in the context of Gibbs sampling in Section 3.4.1, by figure 5.1. We return to this issue in Section 4.3.4, and from now on *re-define* the approximate predictive distribution for record l to

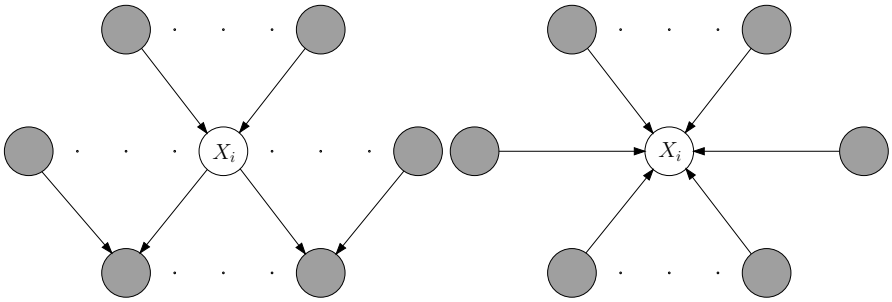

Figure 5.8. New model m' derived from m. All variables in the Markov blanket of X_i in m (*left*) are directed towards X_i in m' (*right*).

be:

$$\Pr(\boldsymbol{U}^l|\boldsymbol{o}^l, m) \stackrel{\text{def}}{=} \prod_{k=1}^{r(l)} \Pr(U_k^l|\boldsymbol{x}_{mb(k)}^l, m)$$

where $\Pr(X_i|\boldsymbol{X}_{mb(i)}, m)$ is the so-called *univariate predictive distribution for* X_i.

The BN decomposition of this predictive distribution according to m is then:

$$\Pr(X_i|\boldsymbol{X}_{mb(i)}, m) \propto \hat{\theta}_{X_i|\boldsymbol{X}_{pa(i)}} \prod_{j \in ch(i)} \hat{\theta}_{X_j|\boldsymbol{X}_{pa(j)}} \tag{5.13}$$

Hence, in order to predict the missing value, we need first compute the estimates $\hat{\theta}_{X_j|\boldsymbol{X}_{pa(j)}}$. To avoid iteration, we could instead do available cases analysis, or apply BC for that purpose. The problem with these approaches is that they in one way or another only consider observations in the data pertaining directly to X_j and $\boldsymbol{X}_{pa(j)}$; they neglect the fact that other variables also influence the estimation when X_j or $\boldsymbol{X}_{pa(j)}$ are (partly) missing. Disregarding this influence effectively means that valuable information is discarded that otherwise could help in making a better and perhaps unbiased prediction of X_i via eq. 5.13.

We propose to change the model m such that the parametrisation of the predictive distribution of X_i remains dependent on the Markov blanket of X_i, but in a way that predictions are less sensitive to slightly inaccurate parameter estimates.

Given the variables in the Markov blanket of m, the univariate predictive distribution based on this new model m' captures no extra assumption about independence compared to the predictive distribution based on m. Specifically, define m' such that $\boldsymbol{X}_{pa(i)}^{m'} = \boldsymbol{X}_{mb(i)}^{m}$ and $\boldsymbol{X}_{ch(i)}^{m'} = \varnothing$,

i.e., extend the parent set of X_i to include *all* variables of the Markov blanket of X_i and remove any children. Figure 5.8 illustrates m' derived from m. We note that an m' is defined for each univariate predictive distribution; m' is only concerned with X_i and the variables in $\boldsymbol{X}_{mb(i)}$ (of m). The independences *actually* holding in m, are disregarded without further ado for predicting missing values:

$$\begin{aligned} \Pr(X_i|\boldsymbol{X}_{pa(i)}, m') &= \theta^{m'}_{X_i|\boldsymbol{X}_{pa(i)}} \\ &\leq_{CI} \Pr(X_i|\boldsymbol{X}_{mb(i)}, m) \propto \hat{\theta}^{m}_{X_i|\boldsymbol{X}_{pa(i)}} \prod_{j \in ch(i)} \hat{\theta}^{m}_{X_j|\boldsymbol{X}_{pa(j)}} \end{aligned} \tag{5.14}$$

Here $\leq_{CI}$ means that the distribution on the left is less restrictive in its conditional independence assumptions compared to the distribution on the right.

Obtaining $\hat{\theta}^{m'}_{X_i|\boldsymbol{X}_{pa(i)}}$ is now done in a way related to available cases analysis, but all variables highly relevant for determining this parameter of the predictive distribution are considered *jointly*. In m' we thus explicitly consider all relevant variables for the parametrisation *together*, in contrast to m where *separate* child-parent variables are considered for estimating the parameters required according to decomposition m.

In a sense m' allows for a rather direct prediction approach; there is no need to perform inference because the most relevant variables for making a prediction are considered jointly. For the predictive distribution based on m, it is indirectly assumed that the correct parametrisation of the actual BN is given *prior* to predicting the missing values; it is by means of (simple) inference that the missing values are predicted, that is, by applying eq. 5.13. However, the only way of estimating these parameters accurately is to have knowledge about the missing values, but of course these missing values are exactly what is subject to prediction.

Note that across equivalent DAG models, vertices have the same Markov blankets, i.e., the Markov blanket of vertices does not change for equivalent DAG models. Thus m' is unique across equivalent DAG models. This entails that the set of predictors is the same no matter which DAG model is considered within an equivalence class. From a statistical point of view this is reassuring, since we explicitly *do not want* to be able to distinguish predictions based on equivalent DAG models.

4.3.3 Parameter estimation

For the parametrisation of the predictive distribution, a *similar cases* approach is used, based on cases where X_i is observed.

As we did for BC, we re-define $n(x_i, \boldsymbol{x}_{pa(i)})$ to pertain to that part of the data where $X_i, \boldsymbol{X}_{pa(i)}$ is fully observed. Hence this function counts the number of occurrences of $x_i, \boldsymbol{x}_{pa(i)}$ in $\boldsymbol{o}$ where child and parents are observed. For the part where X_i is observed but $\boldsymbol{X}_{pa(i)}$ is only partially observed, we define $\tilde{n}(\boldsymbol{x}_{pa(i)}|x_i)$. This function is defined in terms of $match(\boldsymbol{x}^l_{pa(i)}|\boldsymbol{x}_{pa(i)})$ capturing the *degree of match* between $\boldsymbol{x}^l_{pa(i)}$ and $\boldsymbol{x}_{pa(i)}$, when $\boldsymbol{X}^l_{pa(i)}$ is only partially observed in record l. If the observed parents all match with the corresponding values in $\boldsymbol{x}_{pa(i)}$, then it returns the fraction 1/number-of-possible-configurations of the missing parent(s). Formally $match(\boldsymbol{x}^l_{pa(i)}|\boldsymbol{x}_{pa(i)})$ is defined as:

$$\begin{cases} \dfrac{1}{|\Omega_{\boldsymbol{X}^l_{pa(i)} \cap \boldsymbol{U}^l}|} & \text{if } \Big((\boldsymbol{x}_{pa(i)} \cap \boldsymbol{o}^l) = (\boldsymbol{x}^l_{pa(i)} \cap \boldsymbol{o}^l)\Big) \wedge \Big(|\Omega_{\boldsymbol{X}^l_{pa(i)} \cap \boldsymbol{U}^l}| > 0\Big) \\ 0 & \text{otherwise} \end{cases}$$

The function $\tilde{n}(\boldsymbol{x}_{pa(i)}|x_i)$ is now defined in the following way:

$$\tilde{n}(\boldsymbol{x}_{pa(i)}|x_i) = \sum_{l=1}^{c} match(\boldsymbol{x}^l_{pa(i)}|\boldsymbol{x}_{pa(i)}) \cdot I(x^l_i = x_i)$$

Also we define the following total counts:

$$s^*(x_i, \boldsymbol{x}_{pa(i)}) = n(x_i, \boldsymbol{x}_{pa(i)}) + \tilde{n}(\boldsymbol{x}_{pa(i)}|x_i) \tag{5.15}$$

As an example, assume that we have a model $\{X_2, X_3, X_4\} \rightarrow X_1$ with binary variables, and that the following data is given:

	X_1	X_2	X_3	X_4
$\boldsymbol{d}^1$	0	?	1	?
$\boldsymbol{d}^2$	?	0	1	0
$\boldsymbol{d}^3$	1	?	0	1
$\boldsymbol{d}^4$	1	1	0	1

Suppose we require $s^*(X_1 = 1, \boldsymbol{X}_{pa(1)} = \{X_2 = 1, X_3 = 0, X_4 = 1\})$. Record 4 is fully observed, and results in a count of 1 returned by $n(\cdot)$. Any record with $X^1 = 1$ is now considered, and all those records are counted for which the parent set is incomplete, but can be be consistently completed as $\boldsymbol{X}_{pa(1)} = \{X_2 = 1, X_3 = 0, X_4 = 1\}$. In record 3 there is a partial match with the required parent set configuration, and since X_2 has two possible values, 0.5 is returned as the degree of match by $\tilde{n}(\cdot)$. This means that $s^*(X_1 = 1, \boldsymbol{X}_{pa(1)} = \{X_2 = 1, X_3 = 0, X_4 = 1\}) = 1.5$.

For $s^*(X_1 = 0, \boldsymbol{X}_{pa(1)} = \{X_2 = 1, X_3 = 1, X_4 = 0\})$ only record 1 is relevant, and since X_2 and X_4 are missing, there are 4 possible completions, so $s^*(X_1 = 0, \boldsymbol{X}_{pa(1)} = \{X_2 = 1, X_3 = 1, X_4 = 0\}) = 0.25$.

Using the degree of a match, several possible configurations "share the single count" between them which is only fair given the fact that the missing values could have been any of the configurations—the single count is uniformly spread over all possible configurations. When there is a perfect match, $n(\cdot)$ returns a full count, and the other parent configurations receive no share at all.

Gathering the sufficient statistics based on (a subset of) similar observations from the observed records is valid for a broad range of MAR mechanisms, but is not necessarily optimal for arbitrary missing data mechanisms. In contrast to BC and available case analysis, the decomposition according to m' is less sensitive to the actual underlying MAR mechanism because of the larger dependence components captured by considering jointly all variables of the Markov blanket of m.

Obtaining the statistics $s^*(\cdot)$ requires no iteration, and can be done by running through the data sample only once. For each record, all statistics for all vertices X_i are collected simultaneously.

Finally, to estimate $\theta^{m'}_{x_i|\boldsymbol{x}_{pa(i)}}$, we use the following point estimate:

$$\hat{\theta}^{m'}_{x_i|\boldsymbol{x}_{pa(i)}} = \frac{s^*(x_i, \boldsymbol{x}_{pa(i)})}{\sum_{x_i} s^*(x_i, \boldsymbol{x}_{pa(i)})}$$

For complete data, this estimate is the ML estimate, since $\tilde{n}(\cdot)$ will be zero. Note that we may add prior counts $\boldsymbol{\alpha}$ as to obtain the MAP estimates.

4.3.4 Prediction and missing parents

For every single variable in $\boldsymbol{X}$ a predictive distribution can be created in the way discussed in the previous sections and applied in all records where X_i is missing and all parents are observed. However, in some records not all parents of X_i in m' are observed, and consequently the absent parent variables can't be used as predictors. We therefore define the predictive distribution in a slightly different way than in the fully observed parent case. The variables with missing values have to be "summed out" such that only the observed variables act as predictors. For instance, if in record l the variable X^l_i is missing and needs to be predicted, and a subset of predictors $\boldsymbol{V}^l \subseteq \boldsymbol{X}_{pa(i)}$ is missing (so $\boldsymbol{V}^l \subset$

$\boldsymbol{U}^l$), the predictive distribution for X_i^l is $\Pr(X_i^l|\boldsymbol{X}_{pa(i)}^l \setminus \boldsymbol{V}^l, M')$. The parameter for m' is obtained in terms of $s^*(\cdot)$ defined in eq. 5.15 by summing out the missing variables:

$$s^*(x_i^l, \boldsymbol{x}_{pa(i)}^l \setminus \boldsymbol{v}^l) = \sum_{\boldsymbol{v}^l} s^*(x_i^l, \boldsymbol{x}_{pa(i)}^l)$$

This means that when the parent set is not fully observed the parameter estimates are functions of these marginal statistics rather than the "original" statistics.

In summary, we need to traverse the sample once for obtaining the statistics $s^*(\cdot)$. From these statistics we can create the required univariate predictive distributions for both complete and incomplete parent sets.

4.3.5 Predictive quality

As stated in eq. 5.14, the predictive distributions based on m and m' differ in the conditional independence assumptions, but in a way that the distribution based on m' can capture any distribution that m can capture. The predictions we can make for the variables with missing values using the distribution based on m can also be made using the distribution based on m', under the assumption that the parameters are estimated from a sufficiently large data sample.

To estimate $\theta^{m'}_{x_i|\boldsymbol{x}_{pa(i)}}$ we need several "examples" with $\boldsymbol{x}_{pa(i)}$ and x_i observed to get an accurate prediction. If the cardinality of $\boldsymbol{X}_{pa(i)}$ is large there are potentially many different configurations; there may not be enough examples of a particular configuration $\boldsymbol{x}_{pa(i)}$ to make a reliable estimate of $\theta^{m'}_{x_i|\boldsymbol{x}_{pa(i)}}$. Consequently the prediction of X_i may suffer.

The problem is thus that predictions for vertices with a large Markov blanket may be unreliable if the sample size is small. Therefore, we propose to reduce the number of predictors. Instead of taking all vertices of a Markov blanket in m as the parents of X_i in m', we suggest to select as parents the best b variables of the Markov blanket of X_i in m; best in terms of predictive power.

4.3.6 Selecting predictive variables

First we need to distinguish between the vertices of the Markov blanket in terms of *direct dependences* and *induced dependences*. We have that $\boldsymbol{X}_{pa(i)} \not\perp\!\!\!\perp X_i$ and $\boldsymbol{X}_{ch(i)} \not\perp\!\!\!\perp X_i$, i.e., X_i depends on both parents and children; this is a direct dependence because in m we have the serial

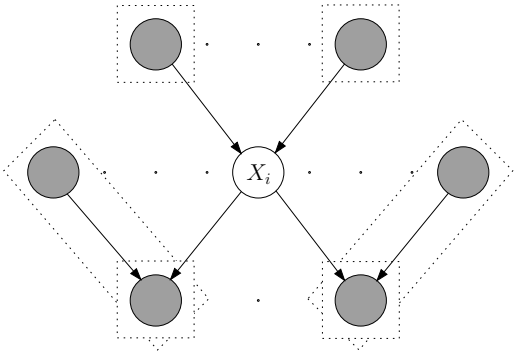

Figure 5.9. The potential predictors of X_i from m. The dotted boxes indicate the predictors that are checked for their predictive quality.

connection $\boldsymbol{X}_{pa(i)} \rightarrow X_i \rightarrow \boldsymbol{X}_{ch(i)}$. The induced dependence says that for every $X_j \in \boldsymbol{X}_{ch(i)}$ we have that $\boldsymbol{X}_{pa(j)} \setminus \{X_i\} \not\perp\!\!\!\perp X_i | X_j$, i.e., X_i depends on the parents of a child of X_i *given* the child. This induced dependence occurs because of the v-connection, $X_i \rightarrow X_j \leftarrow X_k$ with $X_j \in \boldsymbol{X}_{ch(i)}, X_k \in \boldsymbol{X}_{pa(j)}$. This means that if a child of X_i is not chosen to be part of the predictive variables, then in fact we know that the parents of that child are irrelevant as well; we need not consider any of them. Therefore a parent X_k of a child X_j (the collider) is only considered *in conjunction* with X_j as a predictor of X_i. Hence, the set of potential predictors for X_i is:

$$\boldsymbol{\Psi}_i = \boldsymbol{X}_{pa(i)} \cup \boldsymbol{X}_{ch(i)} \cup \{(X_k, X_j) | X_k \in \boldsymbol{X}_{pa(j)} \setminus \{X_i\}, X_j \in \boldsymbol{X}_{ch(i)}\}$$

from which the b best predictors have to be chosen as the parent set of X_i in m'. Figure 5.9 illustrates the different predictors. We thus have to test each predictor, and select the b best.

If we want to check how well a predictor, say X_j, is able to predict X_i, we compute the following probability:

$$\Pr(\boldsymbol{d}_i | \boldsymbol{d}_j, \boldsymbol{X}_{pa(i)} = \{X_j\}) \tag{5.16}$$

where $\boldsymbol{d}_i$ denotes the ith column of the data sample referring to X_i. Hence, given that X_j is parent of X_i (thus act as a predictor in m'), and given the values for the predictor from the data, how well can we predict the values of X_i from the data sample?

The predictive measure in eq. 5.16 coincides with the terms in the recursive factorisation of the marginal likelihood given in eq. 3.18; i.e.,

the marginal likelihood is not used for model learning in the current context, but is now used as a measure of how well a set of observations for X_i can be predicted given another set of observations for X_j.

As mentioned previously, eq. 5.16 has a simple functional form for complete data only. To be more precise, for models like $X_j \rightarrow X_i$, the probability in eq. 5.16 is a simple function of the sufficient statistics of the projection $(\boldsymbol{d}_j, \boldsymbol{d}_i)$ when for every record in the projection it holds that whenever X_i is observed, then X_j is observed too. Obviously, in general we can't guarantee that this is the case. As an alternative we propose to only consider those records where we have observations on X_i as well as on the predictor (in this case X_j). Under the MCAR assumption, this corresponds to an available case analysis of $(\boldsymbol{d}_j, \boldsymbol{d}_i)$, in which case eq. 5.16 is an unbiased probability estimate. Under the more general MAR assumption the probability may be biased when a similar approach is taken. We should however not forget that we are really not interested an entirely correct probability estimate; we merely rank the predictors according to this score, and select the b best predictors. As long as the ranking is not affected in a disastrous way, a biased estimate does not affect the result. Hence, we use eq. 5.16 for arbitrary ignorable missing data mechanisms based on the sufficient statistics collected by an available cases analysis.

Another point which we need to mention is that although we check each predictor on an individual basis, there is no real guarantee that the b best predictors *jointly* are good predictors. The general idea is however that b predictors that score well individually, when taken together produce even better predictions. One may think of alternatives; for instance a "greedy approach" where we maintain a set of predictors, and then add those predictors to the set that increase the predictive score the most until the cardinality of the set is b.

In order to calculate the predictive quality, we need to collect the sufficient statistics from the data sample. This can be done by one traversal through the sample. The statistics thus obtained are not the same as the ones required in the previous section; the b best predictors need to be determined before we can collect $s^*(\cdot)$.

In summary, the so-called MBP (Markov Blanket Predictor) for producing the approximate predictive distribution consists of the steps:

---

1 G-step: In the *generation* step, the univariate predictive distributions are defined:

(a) For each X_i, rank the predictive score of each potential predictor in $\boldsymbol{X}_{mb(i)}$, respecting the induced dependences. Select the top b predictors.
(b) Create m' with the b predictors as parents of X_i, and collect the statistics $s^*(x_i, \boldsymbol{x}_{pa(i)})$.

2 E-step: In the *expectation* step, the expected completion of a record in terms of a count is computed using the univariate predictive distributions. These distributions are parameterised based on $s^*(\cdot)$. If the parent set of a variable that needs prediction is partially missing, sum out the missing parents from $s^*(\cdot)$ first.

4.3.7 Implementation of MBP

Actually using the approximate predictive distribution provided by MBP is part of the "counting-mechanism" employed by the complete data method. When a record is consulted with missing data, instead of counting, the predictive distribution is called, which returns the fraction of a count corresponding to the prediction of the missing value(s) in that particular record.

The pseudo-code in algorithm 3 is an example of how all the statistics are obtained from incomplete data, given a fixed model, i.e., it returns the counts one would obtain in the E-step.

Lines 1–12 correspond to the G-step, where the statistics $s^*(\cdot)$ are collected. These counts are used in the E-step in lines 13–21. Between lines 2–6, the predictors are determined as functions of the Markov blankets. All predictors are scored using the conditional marginal likelihood based on an available cases analysis, and depending on the limit we impose on the maximum number of predictors, the b best predictors are selected.

The lines 14–15 cover the situation where all variables required for obtaining the statistics are observed for a given case. The lines 17–20 covers the situation where some variables are unobserved. These unobserved variables are predicted separately conditional on the predictors from the G-step. If some of the predictors are missing, then they are summed out from the $s^*(\cdot)$ statistics in line 20. Finally, in line 21, the fractional statistics are added.

In the next section we evaluate the MBP algorithm. MBP was implemented in C++ using STL, and was run on a 2 GHz machine under Windows 2000.

Algorithm 3: MBP

Input : b, max. no. of predictors.
Output: $n(\cdot)$, statistics (counts).

for $i \leftarrow 1$ **to** p **do**
  `/* Select predictors for` X_i `and save in` $\boldsymbol{\Psi}_i$ `*/`
  $\boldsymbol{\Psi}_i \leftarrow$`predictors(`$\boldsymbol{X}_{mb(i)}$`)`
  forall Ψ_i **do**
    $q \leftarrow \Pr(\boldsymbol{d}_i|\boldsymbol{d}_{\Psi_i}, \boldsymbol{X}_{pa(i)} = \Psi_i)$
    $\boldsymbol{\mathcal{P}} \leftarrow \boldsymbol{\mathcal{P}} \cup \{(q, \Psi_i)\}$
  $\boldsymbol{\Psi}_i \leftarrow$`selectBest(`$\boldsymbol{\mathcal{P}}, b$`)`
  `/* Get counts` $s^*(X_i, \boldsymbol{\Psi}_i)$ `via` $match$`-function */`
  for $j \leftarrow 1$ **to** c **do**
    if $X_i^j, \boldsymbol{\Psi}_i^j \in \boldsymbol{O}^j$ **then**
      $s^*(x_i^j, \boldsymbol{\psi}_i^j) \leftarrow s^*(x_i^j, \boldsymbol{\psi}_i^j) + 1$
    else
      if $X_i^j \in \boldsymbol{O}^j$ **then**
        forall $\boldsymbol{\psi}_i$ **do**
          $s^*(x_i^j, \boldsymbol{\psi}_i) \leftarrow s^*(x_i^j, \boldsymbol{\psi}_i) + match(\boldsymbol{\psi}_i^j|\boldsymbol{\psi}_i)$
`/* Get counts` $n(X_i, \boldsymbol{X}_{pa(i)})$ `via predictions */`
for $j \leftarrow 1$ **to** c **do**
  if $X_i^j, \boldsymbol{X}_{pa(i)}^j \in \boldsymbol{O}^j$ **then**
    $n(x_i^j, \boldsymbol{x}_{pa(i)}^j) \leftarrow n(x_i^j, \boldsymbol{x}_{pa(i)}^j) + 1$
  else
    $\boldsymbol{\Pi} \leftarrow X_i^j, \boldsymbol{X}_{pa(i)}^j \cap \boldsymbol{U}^j$ `/* Get unobserved variables */`
    forall Π **do**
      $\boldsymbol{V} \leftarrow \boldsymbol{\Psi}_\Pi \cap \boldsymbol{U}^j$ `/* Get unobserved predictors */`
      forall π **do** $\Pr(\pi|\boldsymbol{o}^j) \leftarrow \sum_{\boldsymbol{v}} s^*(\pi, \boldsymbol{\psi}_\Pi^j) / \sum_{\pi,\boldsymbol{v}} s^*(\pi, \boldsymbol{\psi}_\Pi^j)$
    $n(x_i^j, \boldsymbol{x}_{pa(i)}^j) \leftarrow n(x_i^j, \boldsymbol{x}_{pa(i)}^j) + \prod_{\pi \in x_i^j, \boldsymbol{x}_{pa(i)}^j} \Pr(\pi|\boldsymbol{o}^j)$

4.3.8 Parameter estimation

First we compare MBP with standard EM for fitting a BN. We use the same data set as we did with the eMC^4 sampler, about risk factors of coronary heart disease (see Section 3.5.6). The model consists of 6

Table 5.1. Parameter estimates for the model in figure 5.3 for different fractions of missing data according to the mechanism to the left in figure 5.4. A bar indicates negation of the binary variable.

$\hat{\theta}$	**0–10%**		**10–20%**		**20–30%**		**30–40%**		
	EM	**MBP**	**EM**	**MBP**	**EM**	**MBP**	**EM**	**MBP**	**AC**
$\hat{\theta}_{A\|C}$	0.475	0.475	0.467	0.466	0.484	0.484	0.482	0.482	0.477
$\hat{\theta}_{B\|C}$	0.130	0.130	0.132	0.132	0.131	0.131	0.133	0.134	0.122
$\hat{\theta}_{B\|\bar{C}}$	0.290	0.290	0.289	0.289	0.284	0.285	0.282	0.284	0.312
$\hat{\theta}_{C\|A,E}$	0.510	0.510	0.488	0.489	0.509	0.504	0.477	0.482	0.501
$\hat{\theta}_{C\|\bar{A},\bar{E}}$	0.481	0.481	0.469	0.472	0.491	0.490	0.504	0.507	0.545
$\hat{\theta}_{C\|A,\bar{E}}$	0.615	0.613	0.617	0.598	0.598	0.580	0.626	0.588	0.569
$\hat{\theta}_{C\|\bar{A},E}$	0.368	0.369	0.407	0.420	0.392	0.413	0.383	0.417	0.440
$\hat{\theta}_{D\|E}$	0.470	0.470	0.469	0.469	0.480	0.480	0.472	0.473	0.501
$\hat{\theta}_{D\|\bar{E}}$	0.604	0.604	0.603	0.603	0.610	0.610	0.606	0.607	0.577
$\hat{\theta}_{E\|A}$	0.478	0.477	0.473	0.465	0.467	0.460	0.448	0.444	0.458
$\hat{\theta}_{E\|\bar{A}}$	0.622	0.621	0.628	0.620	0.620	0.613	0.585	0.582	0.600
$\hat{\theta}_{F\|B}$	0.163	0.162	0.175	0.174	0.163	0.163	0.154	0.154	0.158
$\hat{\theta}_{F\|\bar{B}}$	0.874	0.875	0.870	0.870	0.871	0.871	0.873	0.874	0.867

binary variables, and is depicted to the left in figure 5.3. Four incomplete sets were generated with 0–10%, 10–20%, 20–30% and 30–40% missing values according to the missing data mechanism to the left in figure 5.4. The probability of non-response of variable X_i conditional on a parent configuration of R_i was selected from the specified interval.

The expected sufficient statistics obtained via MBP were used for estimating the parameters of the model. EM was run until a convergence threshold of 0.001. In table 5.1 the MBP and the EM estimates are shown. MBP requires 3 passes through the sample to estimate the parameters, whereas EM requires several passes, every time performing expensive inference in the BN on a per record basis. As the table suggests, the differences between estimates obtained with EM and MBP are small. At 30–40% missing data, only $\theta_{C|A,\bar{E}}$ and $\theta_{C|\bar{A},E}$ suffer slightly. The reason for this is that A, C and E are all required for determining this parameter (E and A are parents of C), and all 3 variables have incomplete observations as can be seem from the model to the left in figure 5.4.

For 30–40% missing data, we included for comparison, the parameter estimates obtained with available case analysis (column AC). AC relies

heavily on the MCAR assumption. We see that MBP on the other hand can cope with MAR missing data as in this example. MBP is more robust against departures from the MCAR assumption compared to available cases analysis.

Estimating the parameters using MBP is almost instantaneous no matter how large the fraction of missing data is. For EM the number of passes through the data depends on the fraction of missing items; more passes were required for larger fractions of missing items. In this regard we should mention that the convergence rate of EM depends on the fraction of missing information which does not necessarily change dramatically for different fractions of missing items.

4.3.9 Model learning

Next we evaluate MBP in a model selection context. We compare MBP to the SEM implementation of Friedman and Elidan, 2005. For MBP we implemented a greedy search hill-climber as the one described in Kocka and Castelo, 2001. Our hill-climber traverses the essential graph space through repeated covered arc reversals. For MBP we select the 5 best predictors.

We considered two benchmark BNs for the experiments: The ALARM network with 37 vertices and 46 arcs Beinlich et al., 1989 and the Insurance network with 27 vertices and 52 arcs Binder et al., 1997.

For the ALARM network 1000 and 5000 records were sampled. Incomplete sets were generated by applying a missingness mechanism where 18 variables (selected at random) could be missing, half of these according to an MCAR mechanism and the other half according to a MAR mechanism. Three incomplete sets were generated with 0–10%, 10–20% and 20–30% missing values on the 18 variables. For the Insurance network 1000 and 2500 records were sampled, and incomplete sets were generated where 14 variables could be missing, half of them according to a MAR mechanism and the other half according to an MCAR mechanism.

The models learned using SEM and MBP were compared using the marginal likelihood score from 10,000 records sampled from the BN learned (DAG model plus parameter) from the smaller complete data sample (1000 and 5000 for ALARM, 1000 and 2500 for Insurance). Hence, the larger this score, the better a learned model is able to predict the 10,000 records that represent the real underlying distribution. We do not consider the original networks as the golden standards because the relatively small learning samples do not necessarily support the original data generating networks anymore. It is more reasonable to compare to

Table 5.2. Log marginal likelihood score for the ALARM network given 1000 and 5000 records. The score is based on 10,000 records sampled from the BNs learned from the 1000 and 5000 complete records.

% missing	1000 records		5000 records	
	SEM	**MBP**	**SEM**	**MBP**
0–10%	-99973	-98906	-110105	-107173
10–20%	-99824	-98792	-110669	-107012
20–30%	-99849	-98800	-111044	-107617

Table 5.3. Log marginal likelihood score for the Insurance network given 1000 and 2500 records. The score is based on 10,000 record sampled from the BNs learned from the 1000 and 2500 complete records.

% missing	1000 records		2500 records	
	SEM	**MBP**	**SEM**	**MBP**
0–10%	-160251	-156628	-144174	-143189
10–20%	-161129	-156610	-149236	-145711
20–30%	-159087	-158944	-144979	-144887

the models learned from these smaller complete data samples; after all we can't expect to do better than what actually *can* be learned from these smaller complete samples.

In table 5.2 the results for the ALARM network are shown. At no time does MBP produce worse models than SEM from a prediction point of view; in fact the score is better for both 1000 and 5000 records. The same conclusion holds for the Insurance network in table 5.3.

That MBP scores better we partly attribute to the fact that MBP does not get stuck in local optima while traversing DAG space due to covered arc reversals. Also, in contrast to (S)EM, MBP will not get trapped in local optima due to an entirely different approach to solving the missing data problem compared to EM. We observed that the models learned using SEM were more complex compared to models learned with MBP.

Finally we run model selection using MBP on the 3 incomplete data sets that were generated for eMC^4 in Section 3.5.6 with 5–10%, 10–

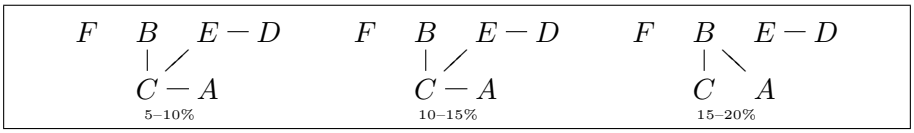

Figure 5.10. The models selected by MBP using the same incomplete data set as for eMC^4.

15% and 15–20% missing components. In figure 5.7 the models selected using SEM are shown, and in figure 5.5 the MAP models from eMC^4 are shown (most frequently visited models on the far left). Figure 5.10 shows the models that were selected using MBP. The models selected for the data with 5–10% and 10–15% missing components are the same that were selected by SEM in figure 5.7, and also the most frequently sampled models using eMC^4. For the data set with 15-20% missing components, the model selected is another one than the model selected by SEM that in turn is different from the most frequently sampled model by eMC^4. However, we see from figure 5.5 that the 4 most frequently visited models are sampled almost equally often (5% to 7%). From a model selection point of view, it hardly matters which model is selected in the end. All in all we may thus conclude that MPB works just as well for model selection as SEM.

4.3.10 Conclusion and discussion

The MBP sampler is an efficient method for generating approximate predictive distributions for incomplete data when learning Bayesian networks. It integrates easily with existing learning approaches for complete data, both for model learning and parameter learning, provided that the methods are functions of the sufficient statistics.

The experiments show that the method works well for both MCAR and the more general MAR missing data mechanisms. It is a very fast method, yet the experiments indicate that the results obtained are comparable to those of EM and SEM. The gain in efficiency is especially noticeable for large data sets where EM and SEM are relatively slow. For massive data sets we believe that MBP provides a real alternative to EM, and in any case is a better choice than available cases analysis and BC. However, we do want to stress that we do not claim that MBP is better or even is able to perform equally well as EM under all circumstances.

On the implementation side, MBP benefits from being very simple to program. In contrast to EM and SEM, no Bayesian networks inference engine is required, which in general is known to be rather tedious to implement.

Chapter 6

CONCLUSION

In this thesis we have discussed and summarised state-of-the-art methods for learning Bayesian networks. In recent years several dissertations have been written about learning Bayesian networks (BNs) from complete data. This thesis picks up where the previous research on BN learning left off. The main contributions of this thesis can be summarised as follows:

- The marginal likelihood was analysed and derived. For model selection, it was shown to have a similar functional form as a penalised likelihood, making explicit the penalty term. In existing literature on BN learning a clear distinction has been made between MDL/BIC/AIC scores on the one hand and the marginal likelihood score on the other due to the functional dissimilarity. In contrast to the penalised likelihood scores, the functional form of the marginal likelihood (which includes gamma functions) often forms an obstacle from an analytical point of view.

  The rewrite also clearly shows the effects that the prior counts have on the marginal likelihood, namely that they primarily determine the regularisation of the models learned.

- Structural EM (SEM) was discussed and by exploiting the marginal likelihood rewrite as a penalised likelihood, it was shown that SEM is not only approximately correct, but exactly. The original derivation of SEM in conjunction with the marginal likelihood relies on different approximations of the marginal likelihood. We have shown that this is not necessary.

- For MCMC learning of BN models, Gibbs sampling and Metropolis-Hastings sampling were combined in the MB-MCMC algorithm. By analysing the marginal likelihood, we were able to identify the edges that strongly depend on each other. By way of blocking those edges, improved mixing of the chain is achieved and this effectively makes the Markov chain less prone to getting stuck for long periods of time in contrast to a single edge MCMC approach. Moreover, there is no added computational burden, compared to existing MCMC model samplers.

- Data Augmentation (DA) was investigated in the context learning BNs, where the Markov blanket was shown to play an important role. Various DA derivatives were proposed for efficient learning of BNs from incomplete data, where the P-step was eliminated. A MCMC model sampler was presented that employs two interleaved non-trivial MCMC samplers, and the efficiency issues were discussed.

- As a solution to the efficiency problems of the interleaved model MCMC sampler, an importance sampling algorithm called eMC^4 was developed that reused previous imputation. The algorithm is based on the observation that MCMC produces correlated samples, and that the Markov chain is visiting the same states several times in a particular region of the state space before moving on to new regions.

- The Markov Blanket Predictor (MBP) was developed as a non-iterative 2-pass algorithm for generating approximate predictive distributions that can be used for both model learning as well as parameter estimation. In identifying the most relevant predictors, attention is restricted to the variables in the Markov blanket. When the Markov blanket is too large, a subset of variables is chosen depending on their ranking defined in terms of the marginal likelihood. The algorithm outperforms existing ad-hoc methods that only consider the parents of variables as predictors, and the results were comparable to those of EM and SEM.

We have shown that learning from incomplete data is more than a trivial extension to existing learning methods for complete data. From a practical point of view we claim that the assumption of complete data is unrealistic. In that respect this thesis paves the road towards a practical approach to learning Bayesian networks from real-life data.

References

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Auto Control*, AC-19:716–723.

Allison, P.D. (2001). *Missing Data*. Number 07-136 in Quantitative Applications in the Social Science. Sage Publications.

Andersson, S., Madigan, D., and Perlman, M. (1997). A characterization of Markov equivalence classes for acyclic digraphs. *Annals of Statistics*, 25:505–541.

Andrieu, C., de Freitas, N., Doucet, A., and Jordan, M.I. (2003). An introduction to MCMC for machine learning. *Machine Learning*, 50(1-2):5–43.

Angelopoulos, N. and Cussens, J. (2001). Markov chain Monte Carlo using trees-based priors on model structure. In Breese, J. and Koller, D., editors, *Proc. of the Conf. on Uncertainty in AI*, pages 16–23.

Beinlich, I.A, Suermondt, H.J, Chavez, R.M, and Cooper, G.F (1989). The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. In *Proc. of the European Conf. on AI in Medicine*.

Binder, J., Koller, D., Russell, S.J., and Kanazawa, K. (1997). Adaptive probabilistic networks with hidden variables. *Machine Learning*, 29:213–244.

Bouckaert, R. (1995). *Bayesian Belief Networks: From Construction to Inference*. PhD thesis, Utrecht University.

Brand, J. P. L. (1999). *Development, Implementation and Evaluation of Muliple Imputation Strategies for the Statistical Analysis of Incomplete Data Sets*. PhD thesis, Erasmus University Rotterdam.

Bromberg, F., Margaritis, D., and Honavar, V. (2006). Efficient markov network structure discovery using independence tests. In *SIAM International conf. on data mining*, pages 141–152.

Buntine, W. (1991). Theory refinemement on Bayesian networks. In D'Ambrosio, B., Smets, P., and Bonissone, P., editors, *Proc. of the Conf. on Uncertainty in AI*.

Carlin, Bradley P. and Louis, Thomas A. (2000). *Bayes and empirical bayes methods for data analysis*. Chapman & Hall/CRC, 2nd edition.

Casella, G. and Berger, R.L. (2002). *Statistical Inference*. Pacific Grove, 2nd edition.

Castelo, R. (2002). *The Discrete Acyclic Digraph Markov Model in Data Mining*. PhD thesis, Utrecht University.

Castelo, R. and Kocka, T. (2003). On inclusion-driven learning of Bayesian networks. *J. of Machine Learning Research*, 4:527–574.

Chickering, D.M. (1995). A transformational characterization of equivalent Bayesian networks. In Besnard, P. and Hanks, S., editors, *Proc. of the Conf. on Uncertainty in AI*, pages 87–98.

Chickering, D.M. (2002a). Learning equivalence classes of Bayesian-network structures. *Journal of Machine Learning Research*, 2:445–498.

Chickering, D.M. (2002b). Optimal structure identification with greedy search. *J. of Machine Learning Research*, 3:507–554.

Chickering, D.M. and Heckerman, D. (1997). Efficient approximations for the marginal likelihood of Bayesian networks with hidden variables. *Machine Learning*, 29:181–212.

Cooper, G. (1990). The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence Journal*, 42:393–405.

Cooper, G. and Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9(4):309–347.

Cowell, R.G. (1998). Mixture reduction via predictive scores. *Statistics and Computing*, 8(2):97–103.

Cowell, R.G., Dawid, A.P., Lauritzen, S.L., and Spiegelhalter, D.J. (1999). *Probabilistic Networks and Expert Systems*. Springer.

Cowell, R.G., Dawid, A.P., and Sebastiani, P. (1995). A comparison of sequential learning methods for incomplete data. *Bayesian Statistics*, 5:533–541.

Cox, D.R. and Hinkley, D.V. (1974). *Theoretical Statistics*. Chapman & Hall.

Dawid, A. P. (1984). Statistical theory: The prequential approach. *J. of Royal Stat. Society (Series A)*, pages 278–292.

Dempster, A.P., Laird, N. M., and Rubin, D.B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *J. of the Royal Statistical Society, Series B*, 34:1–38.

Didelez, V. and Pigeot, I. (1998). Maximum likelihood estimation in graphical models with missing values. *Biometrika*, 85(4):960–966.

Edwards, D. and Havránek, T. (1985). A fast procedure for model search in multidimensional contingency tables. *Biometrika*, 72(2):339–351.

Feller, W. (1970). *An Introduction to Probability Theory and Its Applications*. Wiley.

Friedman, N. (1997). Learning Bayesian networks in the presence of missing values and hidden variables. In *Intl. Conf. on Machine Learning*, pages 125–133.

Friedman, N. (1998). The Bayesian structural EM algorithm. In Cooper, G. F. and Moral, S., editors, *Proc. of the Conf. on Uncertainty in AI*, pages 129–138.

Friedman, N. and Elidan, G. (2005). LibB for Windows/Linux programs 2.1. Can be downloaded at `http://www.cs.huji.ac.il/labs/compbio/LibB`.

Friedman, N. and Koller, D. (2003). Being Bayesian about network structure. A Bayesian approach to structure discovery in Bayesian networks. *Machine Learning*, 50(1–2):95–125.

Gelman, A., Carlin, J.B., Stern, H.S., and Rubin, D. B. (2004). *Bayesian data analysis*. Chapman & Hall/CRC, 2nd edition.

Gelman, A. and Rubin, D.B. (1992). Inference from iterative simulation using multiple sequences. *Statistical Science*, 7(4):457–472.

Geman, S., Bienenstock, E., and Doursat, R. (1992). Neural networks and the bias/variance dilemma. *Neural Computation*, 6:1–58.

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6(6):721–741.

Geweke, J. (1989). Bayesian inference in econometric models using Monte Carlo integration. *Econometrica*, 74(6):1317–1339.

Gilks, W., Richardson, S., and Spiegelhalter, D. (1996). *Markov Chain Monte Carlo in Practice*. Chapman and Hall.

Gillispie, S. and Perlman, M. (2001). Enumerating Markov Equivalence Classes of Acyclic Digraph Models. In *Proc. of the Conf. on Uncertainty in AI*.

Giudici, P. and Green, P. (1999). Decomposable graphical gaussian model determination. *Biometrika*, 86(4):785–801.

Grünwald, P.D., Myung, I.J., and Pitt, M.A. (2005). *Advances in Minimum Description Length: Theory and Applications (Neural Information Processing)*. The MIT Press.

Green, P. (1998). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82:711–732.

Hand, D.J. and Henley, W.E. (1997). Statistical classification methods in consumer credit scoring: A review. *J. of the Royal Stat. Soc., Series A*, 160(3):523–541.

Hastings, W.K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109.

Heckerman, D. (1998). A tutorial on learning with Bayesian networks. In Jordan, M., editor, *Learning in Graphical Models*. MIT Press.

Heckerman, D., Geiger, D., and Chickering, D.M. (1995). Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20:197–243.

Hesterberg, T. (1995). Weighted average importance sampling and defensive mixture distributions. *Technometrics*, 37(2):185–194.

Jensen, C. S., Kong, A., and Kjaerulff, U. (1995). Blocking-Gibbs sampling in very large probabilistic expert systems. *Intl. J. Human-Computer Studies*, pages 647–666.

Jensen, F., Lauritzen, S. L., and Olesen, K. G. (1990). Bayesian updating in causal probabilistic networks by local computations. *Computational Statistics Quarterly*, 4:269–282.

Jensen, F. V. (2001). *Bayesian Networks and Decision Graphs*. Springer-Verlag.

Kennedy, W. and Gentle, J. (1980). *Statistical Computing*. Marcel Dekker, Inc.

Kocka, T. (2001). *Graphical Models: learning and applications*. PhD thesis, Faculty of Informatics and Statistics, University of Economics, Prague.

Kocka, T. and Castelo, R. (2001). Improved learning of Bayesian networks. In Koller, D. and Breese, J., editors, *Proc. of the Conf. on Uncertainty in AI*, pages 269–276.

Larrañaga, P., Poza, M., Yurramendi, Y., Murga, R., and Kuijpers, C. (1996). Structure learning of Bayesian networks by genetic algorithms: A performance analysis of control parameters. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(4):487–493.

Lauritzen, S. L. (1995). The EM algorithm for graphical association models with missing data. *Computational Statistics and Data Analysis*, 19:191–201.

Lauritzen, S. L., Dawid, A. P., Larsen, B., and Leimer, H. (1990). Independence properties of directed Markov fields. *Networks*, 20:491–505.

Lauritzen, S. L. and Spiegelhalter, D. J. (1988). Local computations with probabilities on graphical structures and their applicatins to expert systems. *J. R. Stat. Soc. B*, B(50):127–224.

Lehmann, E.L. and Casella, G. (2001). *Theory of point estimation*. Springer, 2nd edition.

Little, R.J. and Rubin, D.B. (1987). *Statistical analysis with missing data*. Wiley and Sons.

Liu, J.S. (1994). The collapsed Gibbs sampler in Bayesian computations with applications to a gene regulation problem. *J. of the Am. Stat. Assoc.*, 89:958–288.

Liu, J.S., Wong, W.H., and Kong, A. (1994). Covariance structure of the Gibbs sampler with applications to the comparisons of estimators and augmentation schemes. *Biometrika*, 81:27–40.

Madigan, D. and Raftery, A. (1994). Model selection and accounting for model uncertainty in graphical models using Occam's window. *J. of the Am. Stat. Assoc.*, 89:1535–1546.

Madigan, D. and York, J. (1995). Bayesian graphical models for discrete data. *Intl. Statistical Review*, 63:215–232.

McLachlan, G.J. and Krishnan, T. (1997). *The EM algorithm and extensions*. Wiley.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M.N., Teller, A.H., and Teller, E. (1953). Equations of state calculations by fast computing machine. *J. Chem. Phys.*, 21:1087–1091.

Neal, R. (1993). Probabilistic Inference Using Markov Chain Monte Carlo Methods. Technical report, Univ. Toronto.

Nielsen, J., Kocka, T., and Pena, J. (2003). On local optima in learning Bayesian networks. In *Proc. of the Conf. on Uncertainty in AI*, pages 435–442.

Pearl, J. (1986). Fusion, propagation, and structuring in belief networks. *Artificial Intelligence*, 29(3):241–288.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann.

Pearl, J. (2000). *Causality: Models, Reasoning and Inference*. Cambridge Univ. Press.

Pearl, J. and Verma, T. (1987). The logic of representing dependencies by directed graphs. In *Proc. of the Conf. of the American Association of AI*, pages 374–379.

Pelikan, M., Goldberg, D.E., and Cantú-Paz, E. (1999). BOA: The Bayesian optimization algorithm. In *Proc. of the Genetic and Evolutionary Computation Conf.*, volume I, pages 525–532.

Pelikan, M., Goldberg, D.E., Ocenasek, J., and Trebst, S. (2003). Robust and scalable black-box optimization, hierarchy, and Ising spin glasses. Technical Report IlliGAL No. 2003019, Genetic Algorithms Laboratory, University of Illinois.

Ramoni, M. and Sebastiani, P. (1997a). Efficient parameter learning in Bayesian networks from incomplete databases. Technical Report KMI-TR-41, Knowledge Media Institue, The Open University.

Ramoni, M. and Sebastiani, P. (1997b). Learning Bayesian networks from incomplete databases. In Geiger, D. and Shenoy, P., editors, *Proc. of the Conf. on Uncertainty in AI*, pages 401–408.

Ramoni, M. and Sebastiani, P. (1998). Parameter Estimation in Bayesian networks from incomplete databases. *Intelligent Data Analysis Journal*, 2(1).

Ramoni, M. and Sebastiani, P. (2001). Robust learning with missing data. *Machine Learning*, 45(2):147–170.

Riggelsen, C. (2004). Learning Bayesian network parameters from incomplete data using importance sampling. In *European Workshop on Probabilistic Graphical Models*, pages 169–176.

Riggelsen, C. (2005). MCMC learning of Bayesian network models by Markov blanket decomposition. In Gama, J., Camacho, R., Bazdil, P., Jorge, A., and Torgo, L., editors, *European Conf. on Machine Learning*, pages 329–340.

Riggelsen, C. (2006a). Learning Bayesian networks from incomplete data: An efficient method for generating approximate predictive distributions. In *SIAM International conf. on data mining*, pages 130–140.

Riggelsen, C. (2006b). Learning parameters of Bayesian networks from incomplete data via importance sampling. *Intl. J. of Approximate Reasoning*, 42(1-2):69–83.

Riggelsen, C. and Feelders, A. (2005). Learning Bayesian network models from incomplete data using importance sampling. In Cowell, R. G. and Ghahramani, Z., editors, *Proc. of Artificial Intelligence and Statistics*, pages 301–308.

Robert, C.P. and Casella, G. (2002). *Monte Carlo statistical methods*. Springer-Verlag, 3rd edition.

Roberts, G.O. and Sahu, S.K. (1997). Updating schemes, correlation structures, blocking and parametrization for the Gibbs sampler. *J. of the Royal Statistical Society, Series B*, 59:291–317.

Rubin, D.B. (1976). Inference and missing data. *Biometrika*, 63:581–592.

Rubin, D.B. (1987). *Multiple imputation for nonresponse in surveys*. Wiley and Sons.

Schwarz, G. (1978). Estimating dimensions of a model. *Annals of Stats.*, 6(2):461–464.

Seillier-Moiseiwitsch, F., Sweeting, T.J., and Dawid, A.P. (1992). Prequential tests of model fit. *Scand. J. of Statistics*, 19:45–60.

Shafer, J.L. (1997). *Analysis of Incomplete Multivariate Data*, volume 72 of *Monographs on Statistics and Applied Probability*. Chapman & Hall/CRC.

Singh, M. (1997). Learning Bayesian networks from incomplete data. In *American Assoc. for AI*, pages 534–539.

Spiegelhalter, D.J. and Lauritzen, S.L. (1990). Sequential updating of conditional probabilities on directed graphical structures. *Networks*, 20:579–605.

Spirtes, P., Glymour, C., and Scheines, R. (2000). *Causation, Prediction, and Search*. MIT Press. 2nd edition.

Steck, H. and Jaakkola, T. (2002). On the Dirichlet prior and Bayesian regularization. In *Advances in Neural Info. Proc. Systems, 15*.

Tanner, M. and Wong, W. (1987). The calculation of posterior distributions by data augmentation. *J. of the Am. Stat. Assoc.*, 82(398):528–540.

Tierney, L. (1994). Markov chains for exploring posterior distributions. *The Annals of Statistics*, 22:1701–1762.

Titterington, D.M., Smith, A.F.M., and Makov, U.E. (1985). *Statistical analysis of finite mixture distributions*. Wiley.

Verma, T. and Pearl, J. (1990). Equivalence and synthesis of causal models. In *Proc. of the Conf. on Uncertainty in AI*.

Whittaker, J. (1990). *Graphical Models in Applied Multivariate Statistics*. Wiley.

Nederlandse samenvatting

Bayesiaanse netwerken zijn statistische modellen, die de relaties en interacties tussen variabelen kunnen weergeven. Het zijn gerichte grafische modellen, een klasse van statistische modellen die is gedefinieerd door een aantal conditionele onafhankelijkheden tussen variabelen die in een graaf kunnen worden uitgedrukt. Doordat de onderlinge relaties en interacties tussen de variabelen makkelijk af te lezen zijn uit een graaf, is de interpretatie van Bayesiaanse netwerken uitermate aantrekkelijk.

Bayesiaanse netwerken kunnen worden beschouwd als een compacte beschrijving van een kansverdeling over alle variabelen in een domein. Naast de intuïtieve interpreteerbaarheid kunnen Bayesiaanse netwerken dus ook als formalisme voor het redeneren worden gebruikt door de regels van kansrekening toe te passen. Dit is één van de redenen waarom Bayesiaanse netwerken vaak in beslissingsondersteunende systemen worden toegepast.

In dit proefschrift worden methoden besproken voor het leren van Bayesiaanse netwerken uit data. Doordat Bayesiaanse netwerken zeer flexibel zijn en een grote uitdrukkingskracht hebben, hopen we dat het proces dat de data heeft gegenereerd kan worden weergegeven in een Bayesiaans netwerk. Door vervolgens het netwerk te interpreteren kunnen waardevolle inzichten worden verkregen in het proces dat de data heeft gegenereerd.

In dit verband dient de vraag te worden gesteld of het voldoende is om slechts een enkel Bayesiaans netwerk te leren. Doordat we maar over een beperkte hoeveelheid data beschikken, zijn er meer oplossingen mogelijk. Sommige zijn waarschijnlijker dan andere, maar de beste oplossing is niet per se veel beter dan de daarop volgende beste oplossing, enz. De Bayesiaanse statistiek schrijft voor dat we de verdeling over alle mo-

gelijke oplossingen dienen te leren. Om dat te bewerkstelligen moet een beroep worden gedaan op Markov chain Monte Carlo (MCMC) methoden. MCMC-methoden zijn stochastische simulatiemethoden waarmee trekkingen (oplossingen) uit een verdeling kunnen worden gegenereerd. Doorgaans is MCMC-simulatie vanuit een computationeel oogpunt een dure aangelegenheid. In dit proefschift ontwikkelen we methoden voor het efficiënt toepassen van MCMC-methoden voor het leren van Bayesiaanse netwerken.

Het is niet eenvoudig om Bayesiaanse netwerken te leren uit data, vooral niet wanneer de data onvolledig is. Ook hier kan de Bayesiaanse aanpak door middel van MCMC-methoden worden toegepast. In dit opzicht is deze aanpak ook interessant omdat hiermee inzicht wordt verkregen in de toegenomen onzekerheid die te wijten is aan ontbrekende data. We bespreken de moeilijkheden die optreden bij het leren van Bayesiaanse netwerken uit incomplete data en we ontwikkelen hiervoor efficiënte MCMC-methoden.

Ook al heeft de Baysiaans statistische aanpak de voorkeur, bij zeer grote hoeveelheden data zijn MCMC-methoden inefficiënt omdat de data vele malen moet worden doorlopen. Bovendien is de toegevoegde waarde van een Bayesiaanse aanpak bij grote hoeveelheden data vrij klein vergeleken met een aanpak waar maar één enkele oplossing wordt gegenereerd. Bij grote hoeveelheden data is de beste oplossing namelijk veel beter dan de tweede beste oplossing, enz. Het nut van een Bayesiaanse aanpak is daarom gering en kan achterwege worden gelaten. Door slechts een beperkt aantal keren de data te doorlopen kunnen we zeer efficiënte methoden ontwikkelen voor het leren uit grote hoeveelheden data. Deze methoden leveren weliswaar alleen een benadering op, maar vergeleken met exacte methoden zijn de verschillen te verwaarlozen.

Dansk resumé

Baysianske netværk er statistiske modeller som kan illustrere relationer og interaktioner imellem forskellige variabeler. De regnes for såkaldte orienterede grafiske modeller, som tilhører en klasse af statistiske modeller der er karakteriseret ved et antal konditionelle uafhængigheder imellem variabeler, der kan afbilledes i en graf. Relationerne og interaktionerne er nemme at aflæse i grafen, og bla. derfor er Bayesianske netværk ud fra et fotolkningsmæssigt synspunkt fortrinlige til at modelere virkelige systemer med.

Bayesianske netværk kan ses som en kompakt repræsentation af en sandsynlighedsfordeling over variabler fra et bestemt domæne. Det betyder at Bayesianske netværk ikke kun er nemme at fortolke, men tilligemed kan bruges som ræsonnementsværktøj baseret på statistisk inferens og på sandsynlighedsteori. Det er en af grundene til at Bayesianske netværk bruges meget i såkaldte beslutningsstøttesystemer.

I denne afhandling behandler vi forskellige metoder til at lære Bayesianske netværk på fra data. Idet Bayesianske netværk er meget fleksible og har en stor udtrykskraft, er det vores håb, at den proces der har lagt til grund til dataene kan modeleres som et Bayesiansk netværk. Efterfølgende kan netværket så fx fortolkes med hensyn til de relationer og interaktioner der måtte eksistere imellem variablerne, med henblik på at få indsigt i den bagvedliggende proces.

I den forbindelse er spørgsmålet så om det er nok bare at lære ét Bayesiansk netværk og behandle det som om det er det eneste rigtige resultat af læreprocessen. Idet vi kun har rådighed over en begrænset mængde data er der potentielt flere mulige løsninger. Nogen er mere sandsynlige end andre, men det betyder ikke at den bedste løsning nødvendigvis er meget bedre en de 2. bedste løsning, etc. I den såkaldte

Baysianske statistik er man interesseret i hele fordelingen over alle mulige løsninger. I praksis bruges Markov chain Monte Carlo (MCMC) simulation til at realisere en Bayesiansk fremgangsmåde til at lære på. MCMC er en stokastisk simulationsmetode som gør det muligt at trække "resultater" fra en fordeling over fx Bayesianske netværk modeller. Normalt anses MCMC simulering som værende computationelt kostbart. I denne afhandling behandler vi forskellige efficiente MCMC metoder til at lære Bayesianske netværk på.

Det er ikke let at lære Bayesianske netværk fra data, slet ikke hvis vi har at gøre med ufuldstændige data. Manglende observationer og "huller" i dataene betyder at det er endnu svære at lære end med fuldstændige data. Også her er den Bayesiansk statistiske fremgangsmåde interessant, fordi der herved skabes indsigt i den forhøjede usikkerhed som følge af at vi har at gøre med ufuldstændige observationer. Vi diskuterer de problemer der optræder når Bayesianske netværk læres fra ufuldstændige data, og vi udvikler efficiente Bayesianske metoder til at lære på.

Selvom den Bayesianske statistiske fremgangsmåde er at foretrække, med store mængder data er MCMC simulation inefficient fordi dataene skal gennemløbes mange gange. Med store mængder data er den tilføjede værdi ved at bruge en Bayesiansk fremgangsmåde da også minimal sammenlignet med en fremgangsmåde hvor den bedste løsning betragtes for værende det endelige resultat af læreprocessen. Som regel er den bedste løsning mange gange bedre end den 2. bedste løsning, etc., og det er derfor ikke noget større problem at give afkald på den Bayesianske statistik og dermed også MCMC, når vi har store mængder data til vores rådighed. Vi udvikler og diskuterer efficiente metoder til at lære på når vi har at gøre med store datamængder, hvor det kun er nødvendigt at gennemløbe dataene et begrænset antal gange. Resultatet er en løsning der kun er tilnærmelsesvis korrekt. Sammenlignet med metoder der gennemløber dataene mange gange og dermed producerer korrekte resultater, er forskellene dog så minimale at det i praksis ikke betyder noget, slet ikke set i lyset af en hurtigere algoritme.

Curriculum Vitae

Carsten Riggelsen was born on 28th September 1975 in Haderslev, Denmark. He attended public primary and lower secondary school in Toftlund, Denmark, and in 1994 he got his upper secondary school diploma from Tønder Gymnasium, Denmark. The same year he began studying at Aarhus University, Denmark, and in 1998 he continued his studies in Cognitive Artificial Intelligence at the Faculty of Philosophy, Utrecht University, The Netherlands. He obtained his Master's degree in 2002 with the thesis named "Induction of Bayesian Networks with a priori Domain Knowledge" under the supervision of prof.dr.ir. Linda van der Gaag. The same year he started as a Ph.D-student at the Department of Information and Computing Sciences at the same university under the supervision of dr. Ad Feelders and prof.dr. Arno Siebes. In 2006 he completed his Ph.D-thesis.

dissertation series

1998-1	Johan van den Akker (CWI)	DEGAS - An Active, Temporal Database of Autonomous Objects
1998-2	Floris Wiesman (UM)	Information Retrieval by Graphically Browsing Meta-Information
1998-3	Ans Steuten (TUD)	A Contribution to the Linguistic Analysis of Business Conversations within the Language/Action Perspective
1998-4	Dennis Breuker (UM)	Memory versus Search in Games
1998-5	E.W.Oskamp (RUL)	Computerondersteuning bij Straftopmeting
1999-1	Mark Sloof (VU)	Physiology of Quality Change Modelling; Automated modelling of Quality Change of Agricultural Products
1999-2	Rob Potharst (EUR)	Classification using decision trees and neural nets
1999-3	Don Beal (UM)	The Nature of Minimax Search
1999-4	Jacques Penders (UM)	The practical Art of Moving Physical Objects
1999-5	Aldo de Moor (KUB)	Empowering Communities: A Method for the Legitimate User-Driven Specification of Network Information Systems
1999-6	Niek J.E. Wijngaards (VU)	Re-design of compositional systems
1999-7	David Spelt (UT)	Verification support for object database design
1999-8	Jacques H.J. Lenting (UM)	Informed Gambling: Conception and Analysis of a Multi-Agent Mechanism for Discrete Reallocation
2000-1	Frank Niessink (VU)	Perspectives on Improving Software Maintenance
2000-2	Koen Holtman (TUE)	Prototyping of CMS Storage Management
2000-3	Carolien M.T. Metselaar (UVA)	Sociaal-organisatorische gevolgen van kennistechnologie; een procesbenadering en actorperspectief
2000-4	Geert de Haan (VU)	ETAG, A Formal Model of Competence Knowledge for User Interface Design
2000-5	Ruud van der Pol (UM)	Knowledge-based Query Formulation in Information Retrieval
2000-6	Rogier van Eijk (UU)	Programming Languages for Agent Communication
2000-7	Niels Peek (UU)	Decision-theoretic Planning of Clinical Patient Management
2000-8	Veerle Coup (EUR)	Sensitivity Analyis of Decision-Theoretic Networks
2000-9	Florian Waas (CWI)	Image Database Management System Design Considerations, Algorithms and Architecture
2000-11	Jonas Karlsson (CWI)	Scalable Distributed Data Structures for Database Management
2001-1	Silja Renooij (UU)	Qualitative Approaches to Quantifying Probabilistic Networks
2001-2	Koen Hindriks (UU)	Agent Programming Languages: Programming with Mental Models
2001-3	Maarten van Someren (UVA)	Learning as problem solving
2001-4	Evgueni Smirnov (UM)	Conjunctive and Disjunctive Version Spaces with Instance-Based Boundary Sets

2001-5	Jacco van Ossenbruggen (VU)	Processing Structured Hypermedia: A Matter of Style
2001-6	Martijn van Welie (VU)	Task-based User Interface Design
2001-7	Bastiaan Schonhage (VU)	Diva: Architectural Perspectives on Information Visualization
2001-8	Pascal van Eck (VU)	A Compositional Semantic Structure for Multi-Agent Systems Dynamics
2001-9	Pieter Jan 't Hoen (RUL)	Towards Distributed Development of Large Object-Oriented Models, Views of Packages as Classes
2001-10	Maarten Sierhuis (UVA)	Modeling and Simulating Work Practice BRAHMS: a Multiagent Modeling and Simulation Language for Work Practice Analysis and Design
2001-11	Tom M. van Engers (VU)	Knowledge Management: The Role of Mental Models in Business Systems Design
2002-01	Nico Lassing (VU)	Architecture-Level Modifiability Analysis
2002-02	Roelof van Zwol (UT)	Modelling and Searching Web-based Document Collections
2002-03	Henk Ernst Blok (UT)	Database Optimization Aspects for Information Retrieval
2002-04	Juan Roberto Castelo Valdueza (UU)	The Discrete Acyclic Digraph Markov Model in Data Mining
2002-05	Radu Serban (VU)	The Private Cyberspace Modeling Electronic Environments inhabited by Privacy-concerned Agents
2002-06	Laurens Mommers (UL)	Applied Legal Epistemology; Building a Knowledge-Based Ontology of the Legal Domain
2002-07	Peter Boncz (CWI)	Monet: A Next-Generation DBMS Kernel For Query-Intensive Applications
2002-08	Jaap Gordijn (VU)	Value Based Requirements Engineering: Exploring Innovative E-Commerce Ideas
2002-09	Willem-Jan van den Heuvel (KUB)	Integrating Modern Business Applications with Objectified Legacy Systems
2002-10	Brian Sheppard (UM)	Towards Perfect Play of Scrabble
2002-11	Wouter C.A. Wijngaards (VU)	Agent Based Modelling of Dynamics: Biological and Organisational Applications
2002-12	Albrecht Schmidt (UVA)	Processing XML in Database Systems
2002-13	Hongjing Wu (TUE)	A Reference Architecture for Adaptive Hypermedia Applications
2002-14	Wieke de Vries (UU)	Agent Interaction: Abstract Approaches to Modelling, Programming and Verifying Multi-Agent Systems
2002-15	Rik Eshuis (UT)	Semantics and Verification of UML Activity Diagrams for Workflow Modelling
2002-16	Pieter van Langen (VU)	The Anatomy of Design: Foundations, Models and Applications
2002-17	Stefan Manegold (UVA)	Understanding, Modeling, and Improving Main-Memory Database Performance
2003-01	Heiner Stuckenschmidt (VU)	Ontology-Based Information Sharing in Weakly Structured Environments
2003-02	Jan Broersen (VU)	Modal Action Logics for Reasoning About Reactive Systems
2003-03	Martijn Schuemie (TUD)	Human-Computer Interaction and Presence in Virtual Reality Exposure Therapy
2003-04	Milan Petkovic (UT)	Content-Based Video Retrieval Supported by Database Technology

2003-05	Jos Lehmann (UVA)	Causation in Artificial Intelligence and Law - A modelling approach
2003-06	Boris van Schooten (UT)	Development and Specification of Virtual Environments
2003-07	Machiel Jansen (UVA)	Formal Explorations of Knowledge Intensive Tasks
2003-08	Yongping Ran (UM)	Repair Based Scheduling
2003-09	Rens Kortmann (UM)	The Resolution of Visually Guided Behaviour
2003-10	Andreas Lincke (UT)	Some Experimental Studies on the Interaction between Medium, Innovation context and Culture
2003-11	Simon Keizer (UT)	Reasoning under Uncertainty in Natural Language Dialogue using Bayesian Networks
2003-12	Roeland Ordelman (UT)	Dutch Speech Recognition in Multimedia Information Retrieval
2003-13	Jeroen Donkers (UM)	Nosce Hostem - Searching with Opponent Models
2003-14	Stijn Hoppenbrouwers (KUN)	Freezing Language: Conceptualisation Processes across ICT-Supported Organisations
2003-15	Mathijs de Weerdt (TUD)	Plan Merging in Multi-Agent Systems
2003-16	Menzo Windhouwer (CWI)	Feature Grammar Systems - Incremental Maintenance of Indexes to Digital Media Warehouses
2003-17	David Jansen (UT)	Extensions of Statecharts with Probability, Time, and Stochastic Timing
2003-18	Levente Kocsis (UM)	Learning Search Decisions
2004-01	Virginia Dignum (UU)	A Model for Organizational Interaction: Based on Agents, Founded in Logic
2004-02	Lai Xu (UT)	Monitoring Multi-party Contracts for E-business
2004-03	Perry Groot (VU)	A Theoretical and Empirical Analysis of Approximation in Symbolic Problem Solving
2004-04	Chris van Aart (UVA)	Organizational Principles for Multi-Agent Architectures
2004-05	Viara Popova (EUR)	Knowledge Discovery and Monotonicity
2004-06	Bart-Jan Hommes (TUD)	The Evaluation of Business Process Modeling Techniques
2004-07	Elise Boltjes (UM)	Voorbeeldig onderwijs; voorbeeldgestuurd onderwijs, een opstap naar abstract denken, vooral voor meisjes
2004-08	Joop Verbeek (UM)	Politie en de Nieuwe Internationale Informatiemarkt, Grensregionale politiële gegevensuitwisseling en digitale expertise
2004-09	Martin Caminada (VU)	For the Sake of the Argument; Explorations into Argument-based Reasoning
2004-10	Suzanne Kabel (UVA)	Knowledge-rich Indexing of Learning-objects
2004-11	Michel Klein (VU)	Change Management for Distributed Ontologies
2004-12	The Duy Bui (UT)	Creating Emotions and Facial Expressions for Embodied Agents
2004-13	Wojciech Jamroga (UT)	Using Multiple Models of Reality: On Agents who Know how to Play
2004-14	Paul Harrenstein (UU)	Logic in Conflict. Logical Explorations in Strategic Equilibrium
2004-15	Arno Knobbe (UU)	Multi-Relational Data Mining
2004-16	Federico Divina (VU)	Hybrid Genetic Relational Search for Inductive Learning
2004-17	Mark Winands (UM)	Informed Search in Complex Games
2004-18	Vania Bessa Machado (UVA)	Supporting the Construction of Qualitative Knowledge Models

2004-19	Thijs Westerveld (UT)	Using Generative Probabilistic Models for Multimedia Retrieval
2004-20	Madelon Evers (Nyenrode)	Learning from Design: Facilitating Multidisciplinary Design Teams
2005-01	Floor Verdenius (UVA)	Methodological Aspects of Designing Induction-Based Applications
2005-02	Erik van der Werf (UM)	AI techniques for the Game of Go
2005-03	Franc Grootjen (RUN)	A Pragmatic Approach to the Conceptualisation of Language
2005-04	Nirvana Meratnia (UT)	Towards Database Support for Moving Object data
2005-05	Gabriel Infante-Lopez (UVA)	Two-Level Probabilistic Grammars for Natural Language Parsing
2005-06	Pieter Spronck (UM)	Adaptive Game AI
2005-07	Flavius Frasincar (TUE)	Hypermedia Presentation Generation for Semantic Web Information Systems
2005-08	Richard Vdovjak (TUE)	A Model-driven Approach for Building Distributed Ontology-based Web Applications
2005-09	Jeen Broekstra (VU)	Storage, Querying and Inferencing for Semantic Web Languages
2005-10	Anders Bouwer (UVA)	Explaining Behaviour: Using Qualitative Simulation in Interactive Learning Environments
2005-11	Elth Ogston (VU)	Agent Based Matchmaking and Clustering - A Decentralized Approach to Search
2005-12	Csaba Boer (EUR)	Distributed Simulation in Industry
2005-13	Fred Hamburg (UL)	Een Computermodel voor het ondersteunen van euthanasiebeslissingen
2005-14	Borys Omelayenko (VU)	Web-Service configuration on the Semantic Web; Exploring how Semantics meets Pragmatics
2005-15	Tibor Bosse (VU)	Analysis of the Dynamics of Cognitive Processes
2005-16	Joris Graaumans (UU)	Usability of XML Query Languages
2005-17	Boris Shishkov (TUD)	Software Specification Based on Re-usable Business Components
2005-18	Danielle Sent (UU)	Test-selection Strategies for Probabilistic Networks
2005-19	Michel van Dartel (UM)	Situated Representation
2005-20	Cristina Coteanu (UL)	Cyber Consumer Law, State of the Art and Perspectives
2005-21	Wijnand Derks (UT)	Improving Concurrency and Recovery in Database Systems by Exploiting Application Semantics
2006-01	Samuil Angelov (TUE)	Foundations of B2B Electronic Contracting
2006-02	Cristina Chisalita (VU)	Contextual issues in the Design and use of Information Technology in Organizations
2006-03	Noor Christoph (UVA)	The role of Metacognitive Skills in Learning to Solve Problems
2006-04	Marta Sabou (VU)	Building Web Service Ontologies
2006-05	Cees Pierik (UU)	Validation Techniques for Object-Oriented Proof Outlines
2006-06	Ziv Baida (VU)	Software-aided Service Bundling - Intelligent Methods Tools for Graphical Service Modeling
2006-07	Marko Smiljanic (UT)	XML Schema Matching - Balancing Efficiency and Effectiveness by Eeans of Elustering
2006-08	Eelco Herder (UT)	Forward, Back and Home Again - Analyzing User Behavior on the Web

2006-09	Mohamed Wahdan (UM)	Automatic Formulation of the Auditor's Opinion
2006-10	Ronny Siebes (VU)	Semantic Routing in Peer-to-Peer Systems
2006-11	Joeri van Ruth (UT)	Flattening Queries over Nested Data Types
2006-12	Bert Bongers (VU)	Interactivation - Towards an e-cology of People, our Technological Environment, and the Arts
2006-13	Henk-Jan Lebbink (UU)	Dialogue and Decision Games for Information Exchanging Agents
2006-14	Johan Hoorn (VU)	Software Requirements: Update, Upgrade, Redesign - towards a Theory of Requirements Change
2006-15	Rainer Malik (UU)	CONAN: Text Mining in the Biomedical Domain
2006-16	Carsten Riggelsen (UU)	Approximation Methods for Efficient Learning of Bayesian Networks