Department of Computer Science
Series of Publications A
Report A-2009-2

The Most Probable Bayesian Network and Beyond

Tomi Silander

To be presented, with the permission of the Faculty of Science of the University of Helsinki, for public criticism in the University Main Building Auditorium XII on May 30th, at 10 o'clock am.

University of Helsinki
Finland

Contact information

Postal address:
Department of Computer Science
P.O. Box 68 (Gustaf Hällströmin katu 2b)
FI-00014 University of Helsinki
Finland

Email address: postmaster@cs.Helsinki.FI (Internet)

URL: http://www.cs.Helsinki.FI/

Telephone: +358 9 1911

Telefax: +358 9 191 51120

ISSN 1238-8645
ISBN 978-952-10-5523-2 (paperback)
ISBN 978-952-10-5524-9 (PDF)
Computing Reviews (1998) Classification: G.3, H.1.1, H.2.8, I.2.6, I.2.8
Helsinki 2009
Helsinki University Print

The Most Probable Bayesian Network and Beyond

Tomi Silander

Department of Computer Science
P.O. Box 68, FI-00014 University of Helsinki, Finland
Tomi.Silander@cs.Helsinki.FI
http://www.cs.helsinki.fi/tomi.silander/

PhD Thesis, Series of Publications A, Report A-2009-2
Helsinki, April 2009, 50+59 pages
ISSN 1238-8645
ISBN 978-952-10-5523-2 (paperback)
ISBN 978-952-10-5524-9 (PDF)

Abstract

This doctoral dissertation introduces an algorithm for constructing the most probable Bayesian network from data for small domains. The algorithm is used to show that a popular goodness criterion for the Bayesian networks has a severe sensitivity problem. The dissertation then proposes an information theoretic criterion that avoids the problem.

Computing Reviews (1998) Categories and Subject Descriptors:

G.3 [Probability and Statistics]: Multivariate statistics and Statistical software

H.1.1 [Models and Principles]: Systems and Information theory — information theory

H.2.8 [Database Management]: Database applications — data mining

I.2.6 [Artificial Intelligence]: Learning — parameter learning

I.2.8 [Artificial Intelligence]: Problem Solving, Control Methods, and Search — dynamic programming, graph and tree search strategies, and heuristic methods

General Terms:
machine learning, data analysis, Bayesian networks

Additional Key Words and Phrases:
minimum description length, statistics, heuristic search

Preface

I have never been too eager to write this dissertation. Proving myself academically has not motivated me enough, and luckily, there has not been a large amount of pressure to write a doctoral dissertation. If anything, avoiding the status of a PhD has offered me a shield against administrative duties for which a mere PhD student is not qualified. But the times may be changing, and now when I set myself to compose a dissertation, I do it with some reluctance fearing that it may well be just a rationalization of my inability to do so.

This doctoral dissertation consists of five original research papers on Bayesian networks. Together they form a short story about a line of research I have conducted during the last 10 years. As such it is an excerpt of a much larger body of research I have been working on in the Complex Systems Computation research group (CoSCo) at the University of Helsinki. The topic is selected because this particular series of papers carries a convenient storyline or at least a start of a story that post hoc can be made sound coherent.

I have prepended the dissertation with a short introduction to the Bayesian networks. Nowadays, Bayesian networks are so popular that it appears superfluous to write yet another introductory exposition of them. There are plenty of excellent tutorials and undergraduate textbooks around, and the topic is common enough to have a Wikipedia entry of its own. A web-search with words "Bayesian networks introduction" gives a long list of tutorials many written by leading scientists in the field.

An introduction to Bayesian networks must also be a part of thousands of PhD dissertations all over the world, and I will not make an attempt to find a new angle or twist to them. My task is simply to tread the beaten path, and at the same time, introduce the notation used in the papers. I could have chosen a more formal approach for my introduction, but I intuitively chose not to. Early on, my work on Bayesian networks contained an element of teaching this and other Bayesian techniques to educational researchers, so the tone has prevailed.

At the same time, a doctoral dissertation should be a showcase of sophistication of my knowledge on the topic. May that be judged more by the actual research papers since they have been written to the colleagues in the scientific community. In the introductory part, I have concentrated more in providing motivation and insight to (learning of) Bayesian networks, pedagogically cutting corners and avoiding interesting detours when possible. I still hope that even an expert on Bayesian networks may find the introduction enjoyable to read as an example of how someone else in the field thinks about the subject. Some of that has to be read between the lines by noticing what has been included and what has been bluntly omitted.

Acknowledgements

It is customary to acknowledge everybody from my great great ancestors to the clerk at the cafeteria serving my morning coffee. Be they acknowledged since they all play a role in my life that has produced this doctoral dissertation.

I want also to thank the department of computer science at the University of Helsinki for the golden handshake that allowed me to finalize this work. A separate lot of gratitude goes to the pre-examiners of this dissertation, professors Kevin Murphy and Jaakko Hollmén, who have played a concrete role in forming it.

However, the people most directly responsible for academically influencing me and my work are the long time members of the CoSCo research group: Henry Tirri, Petri Myllymäki, Teemu Roos, Petri Kontkanen, Hannes Wettig, and Jussi Lahtinen. Most of my deeper understanding on the matters presented in this dissertation has been created in discussions with these individuals. It is customary to state that all errors in my understanding are “naturally” my own fault, but I am willing to share the blame.

I also want to thank Petri Myllymäki, Teemu Roos, and my wife, Tei Laine, for allocating time to proof read versions of this dissertation. This said, for shortcomings in presentation, there is no one to blame but me.

Contents

Paper I: P. Myllymäki, T. Silander, H. Tirri, and P. Uronen. B-course: A web-based tool for Bayesian and causal data analysis. *International Journal on Artificial Intelligence Tools*, 11(3):369–387, 2002.

Paper II: T. Silander and P. Myllymäki. A simple approach for finding the globally optimal Bayesian network structure. In R. Dechter and T. Richardson, editors, *Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence (UAI-06)*, pages 445–452. AUAI Press, 2006.

Paper III: T. Silander, P. Kontkanen, and P. Myllymäki. On sensitivity of the MAP Bayesian network structure to the equivalent sample size parameter. In R. Parr and L. van der Gaag, editors, *Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence (UAI-07)*, pages 360–367. AUAI Press, 2007.

Paper IV: T. Silander, T. Roos, P. Kontkanen, and P. Myllymäki. Factorized normalized maximum likelihood criterion for learning Bayesian network structures. *Proceedings of the 4th European Workshop on Probabilistic Graphical Models (PGM-08)*, pages 257–264, Hirtshals, Denmark, 2008.

Paper V: T. Silander, T. Roos, and P. Myllymäki. Locally minimax optimal predictive modeling with Bayesian networks. In D. van Dyk and M. Welling, editors, *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS-09), Volume 5 of JMLR: W&CP 5*, pages 504–511, Clearwater Beach, Florida, USA, 2009.

For more information about these papers, see Chapter 5.

Part I

Overview of the theory of Bayesian networks

Chapter 1

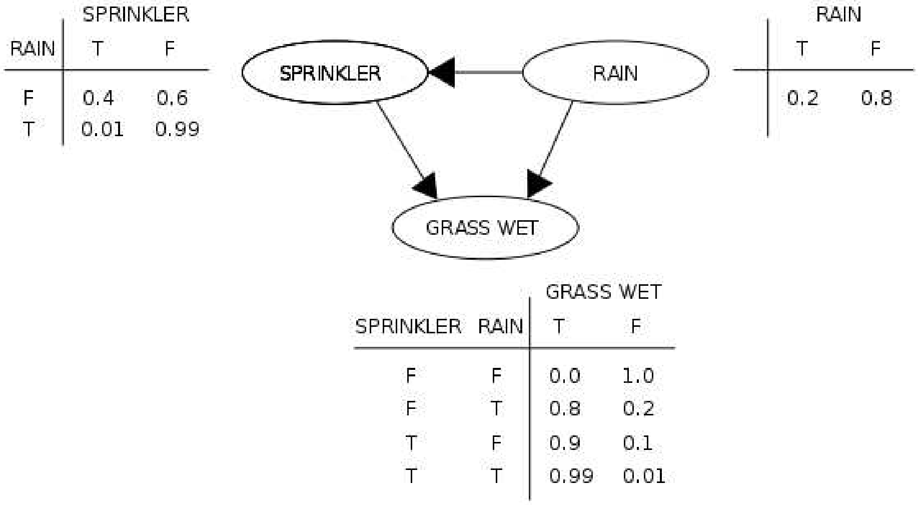
Introduction

> *"It would be entirely superfluous to enumerate how many and how great the advantages of this instrument are on land and at sea. But having dismissed earthly things, I applied myself to explorations of the heavens."*
>
> Galileo Galilei, *Sidereus Nuncius*, 1610.

Bayesian networks [58] have a history of over 20 years now. Their appearance is that of network diagrams that are ubiquitous in many fields of science and humanities (Figure 1.1). In particular, the causal flavour of Bayesian networks [74, 59], in which nodes represent states and arrows represent causal influence, is probably too simple to be assigned a single inventor. In statistics the predecessors of these kind of models are usually stated to be path diagrams [82, 83] and structural equation models [82, 28, 72]. The term "Bayesian belief network" was coined by Judea Pearl [57] and made popular by his 1988 "black book", *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, which is also author's first contact to the Bayesian networks.

Bayesian networks are both intuitive theories about the domain of interest and, at the same time, mathematically specified objects that allow prediction, generalization, and planning. These characteristics make them widely applicable. For example, the web-site of the commercial Bayesian network tool HUGIN[1] [2] lists over twenty different projects in finance, medicine, industry, robotics, food safety, etc. in which Bayesian networks have been used. The methods for automatically constructing Bayesian networks from the data further widens their prospects. Academically, the

[1] `http://www.hugin.com/`



	SPRINKLER	
RAIN	T	F
F	0.4	0.6
T	0.01	0.99

RAIN	
T	F
0.2	0.8

		GRASS WET	
SPRINKLER	RAIN	T	F
F	F	0.0	1.0
F	T	0.8	0.2
T	F	0.9	0.1
T	T	0.99	0.01

Figure 1.1: A causal Bayesian network pictured in Wikipedia (Dec 4, 2008). The network encodes the idea that rain has effect on whether the sprinkler is on or off, and the rain and the sprinkler both affect the wetness of the grass. The picture also contains probability tables that further specify these effects. For example, there is 20% chance of rain (RAIN = T(rue) : 0.2), and if the sprinkler is on, but it does not rain, the grass is wet with probability 0.9.

intuitive nature of Bayesian networks makes them an interesting case study in developing principled ways of using data to construct complex models.

The series of research papers in the second part of this dissertation reflects this continuum from practicalities to more scholarly issues. The first two papers describe tools and algorithms for learning Bayesian networks from data. The rest of the papers use these tools to study the principles for constructing good Bayesian networks. The results of these studies may then be used to improve our network construction capabilities.

Maybe a word about the word “Bayesian” in “Bayesian networks” since it often raises questions, concerns, and confusion. The word “Bayesian” in “Bayesian networks” can be considered a misnomer. Bayesianism is a certain way (in reality, a family of slightly different ways) to give semantic interpretation to the concept of probability. It tries to define what it means to state that the probability of something is, say 0.72. For a Bayesian, the probability of a statement describes how strongly he/she believes in that statement. This interpretation has consequences on how the probability theory can or should be applied for modelling the real world. Naturally, as a branch of mathematics, the probability theory itself does not dictate how it should be correctly applied, but the issue lies outside of mathematics. There is nothing Bayesian in Bayesian networks as such; dependence networks would be a better term. Being statistical models, Bayesian networks can (and will in this work) be used following statistical conventions that are Bayesian, but that is totally another matter, and the coincidence of these words is a source of great confusion for those not initiated in the philosophy of statistics.

When the term Bayesian network was coined, statistics and data analysis were still considered to be very separate from artificial intelligence (AI), and Bayesian networks were mainly a knowledge representation formalism in AI. The word "Bayesian" was introduced to emphasize the subjective nature of this knowledge, conditioning as a method to update probabilities, and the distinction (made by Thomas Bayes) of causal and evidential models of reasoning. Only later did part of AI move very close to statistics, and the choice of words became confusing.

Nowadays, Bayesian networks are seen as one member in a much larger family of graphical models [80, 46]. The following overview of the theory of Bayesian networks is by no means a comprehensive treatise of the topic. It concentrates on the issues that appear in the research papers in the second part of this work. Therefore, there are several topics that have been omitted. Notable omissions of this kind are the algorithms for efficient inference in Bayesian networks [47, 21, 18] and the exact theory of independence relations [19, 60, 46]. For an interested reader there are tutorials [32], books [58, 51, 68, 9, 16, 39], and the web abound of introductions and even video lectures[2].

The rest of this introduction to Bayesian networks is structured as follows. In Chapter 2 we will first introduce Bayesian networks, their motivation, and the notation needed to treat them as mathematical objects. In Chapter 3 we will concentrate on the topic of learning Bayesian networks automatically from the data, which is the main concern of the dissertation. In Chapter 4 we will then briefly discuss some aspects of the theory of minimum description length (MDL), since the solutions we offer in research papers IV and V to the problems mentioned in Chapter 3 and in research paper III, are based on the MDL principle. Finally, in Chapter 5, we offer a glimpse to the background and the main results of the research papers appearing in Part II of this dissertation.

[2] `http://videolectures.net/kdd07_neapolitan_lbn/`

Chapter 2

Bayesian networks

2.1 Bayesian networks as knowledge representation

For a long time, logic was the primary knowledge representation language in artificial intelligence [48, 49, 78]. While well suited for simulated closed world scenarios, its application in real world situations proved challenging. The fall of rule-based expert systems and the apparent failure of Japan's Fifth Generation Computer Systems, largely built on parallel logic programming, called for a new paradigm [24].

Logic	Probability
Dropping the plate breaks it except when the plate is made of steel or such, or the floor is very soft, or somebody catches the plate before it hits the ground, or we are not in the gravity field, or ...	Dropping the plate breaks it 95% of the time.

Figure 2.1: The qualification problem circumvented by using probability.

In its rigidity, logic can only derive truth from truths, and the lack of notion of mere plausibility makes it hard to express facts that, while not necessarily true by logic, are still true in normal circumstances. The freedom of not specifying every possible exception to a rule, but only quantifying the uncertainty by a single real number, a probability, yields a more realistic modelling language, (for an example, see Figure 2.1). At its extreme, when using only probabilities 0 and 1, this language coincides with logic. Therefore, probability can be seen as an extended logic [38] that, governed by the laws of probability theory, allows inferences that yield consequences together with estimates of their plausibility.

Rule 1:	Adding strawberries to food makes it taste better.	(5%)
Rule 2:	Adding mustard to food makes it taste better.	(10%)
So how about	Adding both strawberries and mustard to food makes it taste better.	(??%)

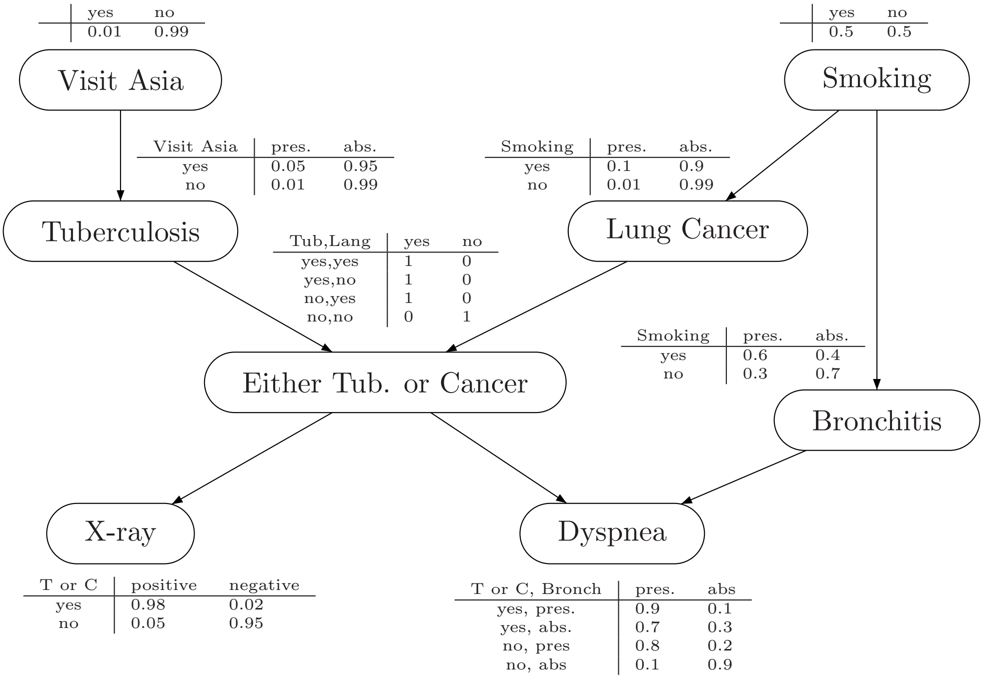
Figure 2.2: Problem of combining evidence based only in certainty factors.

Use of probabilities for knowledge representation has problems of its own. While probability theory has a principled answer to combining correlated evidence [30] (something that plagued the attempts to couple logical rules with so called certainty factors, see Figure 2.2), it does so with an additional cost of requiring specification of joint probabilities of events, i.e, one has to be able provide the probability to all the possible combinations of events that may happen in the domain to be modelled. This requirement makes the naive application of probability theory for knowledge representation infeasible; see Figure 2.4 for an example.

Bayesian networks try to remedy the situation by structuring the joint distribution of the domain into smaller interconnected parts (Figure 2.3). The more compact representation also allows more efficient inference, i.e., calculation of conditional probabilities that measure the plausibility of unknown things in the light of current observations.

In its most robust realization, this structuring conforms to causal mechanisms of the domain yielding a graph of things connected by causal links, so that the probabilities required to measure the causal connections are easier for humans to assess. Causal Bayesian networks do not define the joint probability distribution only on a static domain, but they also specify how the world would react if some of its processes were altered by an external force [74]. This makes it possible to use Bayesian networks for planning and explanation.

While this study does not directly deal with causal Bayesian networks, much of the motivation for the work, and Bayesian networks in general, stems from the realms of causality [59].



	yes	no
	0.01	0.99

	yes	no
	0.5	0.5

Visit Asia	pres.	abs.
yes	0.05	0.95
no	0.01	0.99

Smoking	pres.	abs.
yes	0.1	0.9
no	0.01	0.99

Tub,Lang	yes	no
yes,yes	1	0
yes,no	1	0
no,yes	1	0
no,no	0	1

Smoking	pres.	abs.
yes	0.6	0.4
no	0.3	0.7

T or C	positive	negative
yes	0.98	0.02
no	0.05	0.95

T or C, Bronch	pres.	abs
yes, pres.	0.9	0.1
yes, abs.	0.7	0.3
no, pres	0.8	0.2
no, abs	0.1	0.9

Figure 2.3: A famous example of a causal Bayesian network presenting reasons for shortness-of-breath (dyspnoea) [47]. This presentation is a huge improvement of the naive way of providing the same joint probability distribution (see Figure 2.4).

2.2 Bayesian networks as joint probability distributions

We will now introduce the mathematical notation for defining Bayesian networks as joint probability distributions. The reader is advised to refer to Figure 2.5 to place the notation into the context of a graphical structure.

A Bayesian network defines a joint probability distribution for a vector valued (or multivariate) random variable. In this dissertation we will only consider finite domains in which each coordinate X_i of an n-dimensional random vector $X = (X_1, \ldots, X_n)$ has a finite number of values that, without loss of generality, can be assumed to be $1, \ldots, r_i$.

A Bayesian network consists of two parts: a qualitative part (or structure) that can presented as a directed acyclic graph (DAG), and a quantitative part (or parameters) that further specify the dependence relations

Asia	Smoke	Tuberc.	L. Cancer	Bronchitis	X-ray	Dyspnoea	probability
yes	yes	present	present	present	pos	present	0.00013230
yes	yes	present	present	present	pos	absent	0.00001470
yes	yes	present	present	present	neg	present	0.00000270
yes	yes	present	present	present	neg	absent	0.00000030
yes	yes	present	present	absent	pos	present	0.00006860
yes	yes	present	present	absent	pos	absent	0.00002940
yes	yes	present	present	absent	neg	present	0.00000140
yes	yes	present	present	absent	neg	absent	0.00000060
yes	yes	present	absent	present	pos	present	0.00119070
yes	yes	present	absent	present	pos	absent	0.00013230
yes	yes	present	absent	present	neg	present	0.00002430
yes	yes	present	absent	present	neg	absent	0.00000270
yes	yes	present	absent	absent	pos	present	0.00061740
yes	yes	present	absent	absent	pos	absent	0.00026460
yes	yes	present	absent	absent	neg	present	0.00001260
yes	yes	present	absent	absent	neg	absent	0.00000540
yes	yes	absent	present	present	pos	present	0.00000661
yes	yes	absent	present	present	pos	absent	0.00000073
yes	yes	absent	present	present	neg	present	0.00000013
... and there are still 109 variable configurations to go ...							...
... and there are still 108 variable configurations to go ...							...

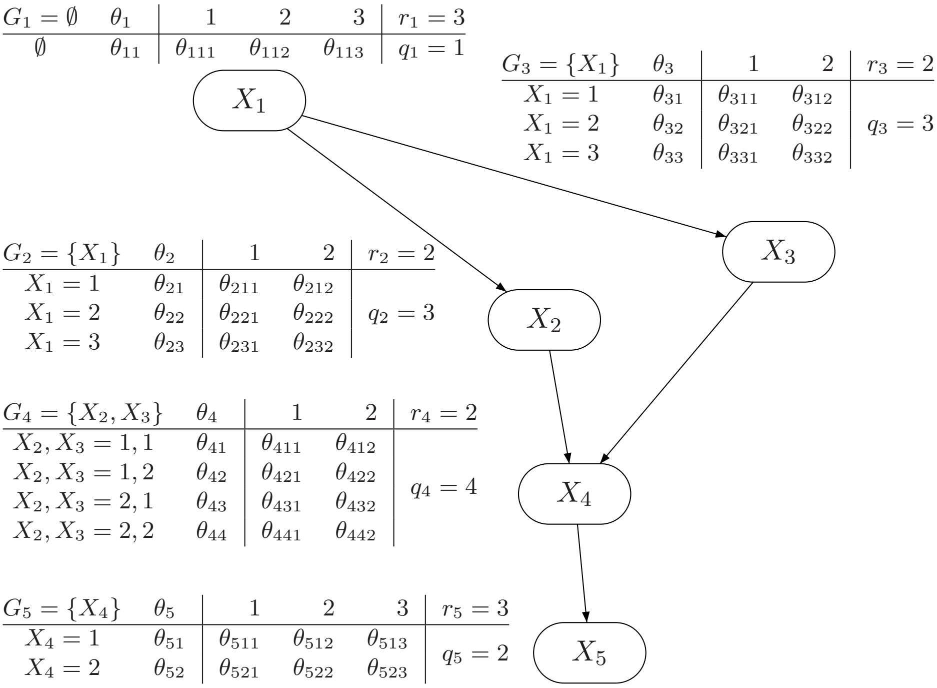
Figure 2.4: First 19 out 128 probabilities needed for a naive specification of the joint probability distribution of the Asia-domain (Figure 2.3). Each added variable at least doubles the size of table.

defined by the structure. We will often denote the structure with a capital letter G (for graph) and the parameters with a Greek letter theta θ[1].

The structure G for an n-dimensional random vector X has exactly one node per each coordinate (also called attribute or variable) of X, and therefore, we often use words "node", "variable" and "attribute" interchangeably. In particular, we often say "node X_i" when we refer to a node that corresponds to the variable X_i.

We code the structure of a Bayesian network as a vector $G = (G_1, \ldots, G_n)$ in which each coordinate G_i denotes a set of those nodes from which there are arcs to node X_i. The set G_i is often called the *parents* of node X_i and the set $G_i \cup \{X_i\}$ the *family* of X_i. Due to the acyclicity requirement, not all vectors of node subsets are valid Bayesian network structures. G_i is an empty set if X_i does not have any parents.

[1] It would be more appropriate to write θ_G since the parameters needed to quantify a structure depend on the structure, but this is usually omitted, i.e., understood from the context.



$G_1 = \emptyset$	θ_1	1	2	3	$r_1 = 3$
$\emptyset$	θ_{11}	θ_{111}	θ_{112}	θ_{113}	$q_1 = 1$

$G_3 = \{X_1\}$	θ_3	1	2	$r_3 = 2$
$X_1 = 1$	θ_{31}	θ_{311}	θ_{312}	$q_3 = 3$
$X_1 = 2$	θ_{32}	θ_{321}	θ_{322}	
$X_1 = 3$	θ_{33}	θ_{331}	θ_{332}	

$G_2 = \{X_1\}$	θ_2	1	2	$r_2 = 2$
$X_1 = 1$	θ_{21}	θ_{211}	θ_{212}	$q_2 = 3$
$X_1 = 2$	θ_{22}	θ_{221}	θ_{222}	
$X_1 = 3$	θ_{23}	θ_{231}	θ_{232}	

$G_4 = \{X_2, X_3\}$	θ_4	1	2	$r_4 = 2$
$X_2, X_3 = 1, 1$	θ_{41}	θ_{411}	θ_{412}	$q_4 = 4$
$X_2, X_3 = 1, 2$	θ_{42}	θ_{421}	θ_{422}	
$X_2, X_3 = 2, 1$	θ_{43}	θ_{431}	θ_{432}	
$X_2, X_3 = 2, 2$	θ_{44}	θ_{441}	θ_{442}	

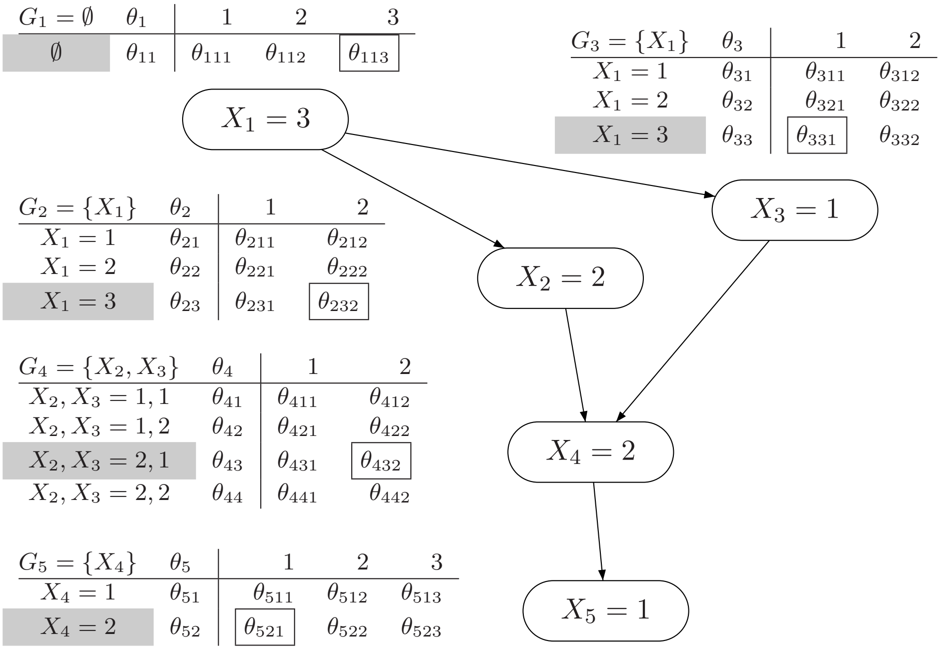
$G_5 = \{X_4\}$	θ_5	1	2	3	$r_5 = 3$
$X_4 = 1$	θ_{51}	θ_{511}	θ_{512}	θ_{513}	$q_5 = 2$
$X_4 = 2$	θ_{52}	θ_{521}	θ_{522}	θ_{523}	

Figure 2.5: A Bayesian network for variables $X = (X_1, X_2, X_3, X_4, X_5)$: $n = 5$, $G = (\{\}, \{X_1\}, \{X_1\}, \{X_2, X_3\}, \{X_4\})$.

Parameters θ follow the structure of the network G. Associated with each set of parents G_i is a table θ_i of parameters that define conditional probability distributions $P(X_i \mid G_i, \theta)$. To this end, the possible values of G_i (often called parent configurations) are enumerated from 1 to q_i ($q_i = \prod_{X_i \in G_i} r_i$), and for each $j \in \{1, \ldots, q_i\}$, there is an r_i-dimensional parameter vector $\theta_{ij} = (\theta_{ij1}, \ldots, \theta_{ijr_i})$ defining the conditional probability $P(X_i = k \mid G_i = j, \theta_i) = \theta_{ijk}$. If G_i is an empty set, we define $q_i = 1$. In order to define a proper conditional probability distribution, the parameters θ_{ijk} must belong to the closed unit interval $[0, 1]$, and for each i and j, the sum $\sum_{k=1}^{r_i} \theta_{ijk}$ must add up to 1.

With this motherload of notation, we can now define the probability of an n-dimensional random vector. Given a Bayesian network $B = (G, \theta)$ the probability of a vector $X = (X_1, \ldots, X_n)$ can be defined as

$$P(X \mid B) = \prod_{i=1}^{n} P(X_i \mid G_i = X_{G_i}, \theta_i) = \prod_{i=1}^{n} \theta_{ij_ik_i}, \tag{2.1}$$



$G_1 = \emptyset$	θ_1	1	2	3
$\emptyset$	θ_{11}	θ_{111}	θ_{112}	$\boxed{\theta_{113}}$

$G_3 = \{X_1\}$	θ_3	1	2
$X_1 = 1$	θ_{31}	θ_{311}	θ_{312}
$X_1 = 2$	θ_{32}	θ_{321}	θ_{322}
$X_1 = 3$	θ_{33}	$\boxed{\theta_{331}}$	θ_{332}

$G_2 = \{X_1\}$	θ_2	1	2
$X_1 = 1$	θ_{21}	θ_{211}	θ_{212}
$X_1 = 2$	θ_{22}	θ_{221}	θ_{222}
$X_1 = 3$	θ_{23}	θ_{231}	$\boxed{\theta_{232}}$

$G_4 = \{X_2, X_3\}$	θ_4	1	2
$X_2, X_3 = 1, 1$	θ_{41}	θ_{411}	θ_{412}
$X_2, X_3 = 1, 2$	θ_{42}	θ_{421}	θ_{422}
$X_2, X_3 = 2, 1$	θ_{43}	θ_{431}	$\boxed{\theta_{432}}$
$X_2, X_3 = 2, 2$	θ_{44}	θ_{441}	θ_{442}

$G_5 = \{X_4\}$	θ_5	1	2	3
$X_4 = 1$	θ_{51}	θ_{511}	θ_{512}	θ_{513}
$X_4 = 2$	θ_{52}	$\boxed{\theta_{521}}$	θ_{522}	θ_{523}

Figure 2.6: $P(X = (3, 2, 1, 2, 1) \mid G, \theta) = \theta_{113}\theta_{232}\theta_{331}\theta_{432}\theta_{521}$.

where j_i denotes the index of the configuration of variables G_i found in X and the k_i denotes the value of X_i; (see Figure 2.6 for an example). That this formula really defines a probability distribution for X is relatively easy to see. The defined probabilities clearly lie in a unit interval $[0, 1]$. An easy way to see that the probabilities do indeed sum to one is to use mathematical induction for the number of variables n. For a single variable the summation clearly holds. For larger n, both the summation over all the variables and the product (2.1) within the sum can be carried out in the order in which the last variable has no children. The summation over the last variable can now be moved over the other variables in the product, and since the last sum equals 1.0, we end up with the sum of probabilities for a Bayesian network with $n - 1$ variables.

2.3 Bayesian networks as generative models

Since a Bayesian network defines the probability of a data vector, it can be used as a data generating machine. To sample a data vector from a

```
**Algorithm 1**: Gendata($B$,*topolorder*):
  **input**  : Bayesian network $B = (G, \theta)$,
               topological ordering *topolorder* of indices {1 ... n} by $G$
  **output**: data vector $X$
  n ← length($G$)
  $X$ ← vector of n numbers all -1
  **for** $i$ **in** *topolorder* **do**
      $j \leftarrow X_{G_i}$
      $X_i \leftarrow$ random_sample_by($\theta_{ij}$)
  **end**
  **return** $X$
```

Bayesian network, one may proceed by generating its coordinates in topological order, i.e., in any order that confirms with the partial ordering of the variables induced by the network structure. The ordering guarantees that the parents are sampled before children, so that it is always easy to generate the variable X_i by the probability distribution $P(X_i \mid G_i = X_{G_i}, \theta_i)$ that is readily available in a network. The Algorithm 1 features pseudocode for generating a random vector from a Bayesian network using this *ancestral sampling* scheme [7]. The algorithm assumes a function "random_sample_by" that generates a single value from a discrete distribution.

By generating N n-dimensional data vectors independently from a Bayesian network B, we can generate an $N \times n$ data matrix D in which each row d^t is a data vector generated from the B. It turns out to be useful to introduce a notation for certain parts of such a data matrix. We often want to select rows of the data matrix by certain criteria. We then write the selection criterion as a superscript of the data matrix D. For example, $D^{G_i=j}$ denotes those rows of D where the variables of G_i have the j^{th} value configuration. We reserve the particular notation of superscripting the D by an integer t (like D^t) for denoting the first t rows of the data matrix D. If we further want to select certain columns of these rows, we denote the columns by subscripting D with the corresponding variable set. As a shorthand, we write $D_{\{X_i\}} = D_i$. For example, $D_i^{G_i=j}$ selects the i^{th} column of the rows $D^{G_i=j}$.

Assuming that the data vectors are independent, a Bayesian network defines the probability of a data matrix D simply by the product

$$P(D \mid B) = \prod_{t=1}^{N} P(d^t \mid B), \tag{2.2}$$

where d^t denotes the t^{th} row of matrix D.

If we insert the Equation (2.1) into Equation (2.2), and then regroup the terms, we can express the probability of the data D using the counts N_{ijk} that tally how many times the value k of the variable X_i appears together with the parent configuration j in a data matrix D:

$$P(D \mid B) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \theta_{ijk}^{N_{ijk}}. \tag{2.3}$$

The counts N_{ijk} (i.e., the number of rows in $D^{X_i=k,G_i=j}$) will play a central role in the theoretical development in the next chapter. In a way, these counts contain all the information a Bayesian network structure can extract from a data sample. In particular, if there are two data matrices D and D' that have the same "sufficient statistics" N_{ijk}, the probabilities of these data matrices are equal.

Chapter 3

Learning Bayesian networks

Although it is easy to generate data from a Bayesian network, the inverse problem is involved. If we have an $N \times n$ data matrix D, and we assume that it was generated by a Bayesian network with n nodes, what was that Bayesian network like? This innocent looking question resembles the scientific inquiry, since it equals inducing a model of the world from observations [33].

The assumption that data was indeed generated by a Bayesian network of n variables is a problematic one. Often the data is a product of constantly changing world that is not naturally described by a single Bayesian network. It is also common that, even if the data vector contains n variables, the actual process of generating the data contains additional factors that are not recorded in our data.

Under the assumption that the data was generated by a Bayesian network, the problem of learning the network is usually divided into two subproblems: learning the structure of a Bayesian network and learning the parameters for it. The latter problem is usually considered easier [52].

The amount of data we have at hand is also a significant factor. In multidimensional discrete domains, the number of possible data vectors is usually huge since it grows exponentially with the number of variables. Therefore, in most practical cases, our data set contains only a tiny fraction of all possible data vectors we “could” have observed. The situation is very different from simple univariate situations, say tossing a single coin, where we usually observe every possible value several times. For this reason, many classical statistical procedures cannot be sensibly applied.

3.1 Learning parameters for Bayesian networks

When learning the parameters for a Bayesian network, we assume that we have a complete data matrix D and that we know the structure G of the network that generated the data. Even then, the problem of learning parameters for a Bayesian network is not precisely defined, but one of the basic questions is "what were the parameter values in the network that generated this data?" We will address this problem first. There is another, related problem of finding the parameters with a good predictive capability, which we will address subsequently.

3.1.1 The maximum likelihood parameters

The classical statistical approach to finding the data generating parameters is the method of maximum likelihood (ML) which calls for finding the parameter values that give the data D at least as large a probability as any other parameter values do.

In the case of Bayesian networks, this is a rather simple task. Due to the very modular structure of the likelihood function (2.3), the maximization task reduces to the maximization of the likelihood function of a single multinomial variable. For those parameters θ_{ij} for which there is at least one data vector with the configuration j in variables G_i, the maximizing parameters are simply the relative frequencies $\hat{\theta}_{ijk} = \frac{N_{ijk}}{\sum_{k'=1}^{r_i} N_{ijk'}}$; (see Figure 3.1 for an example). The parameters corresponding to the parent configurations that do not appear in the data, do not contribute to the probability of the data, so they can be set to any valid values. To fix the maximum likelihood parameters, we adopt the convention of setting those parameters according to the uniform distribution $\hat{\theta}_{ijk} = \frac{1}{r_i}$. This choice is somewhat arbitrary, and other conventions like $\hat{\theta}_{ijk} = \frac{N_{ik}}{N}$, where $N_{ik} = \sum_{j=1}^{q_i} N_{ijk}$, could be justified as well. In the future we will not have any use for these unsupported maximum likelihood parameters, so for this work, the choice does not really matter.

3.1.2 Bayesian learning of the parameters

Bayesian statistics spares us from some of the problems of the maximum likelihood parameters[1]. Unlike classical statisticians (i.e, so called frequentists), Bayesians handle the uncertainty about parameters by treating them

[1] These problems will be discussed later in section 3.1.3.

as random variables. This leads to a conditional distribution for the parameters, which can be calculated by the Bayes' theorem

$$P(\Theta \mid G, D) = \frac{P(D \mid \Theta, G)P(\Theta \mid G)}{P(D \mid G)}.$$

In order to calculate this *posterior distribution*, a Bayesian is required to specify a *prior distribution* $P(\Theta \mid G)$. To make the task of specifying the prior manageable, it is common to deploy a set of (dubious) tricks that yield the task more bearable or even trivial. First, the parameter vectors Θ_{ij} are assumed to be independent of each other, so we may assign them a probability distribution one by one. Second, the form of the prior distribution is also selected to be conjugate to the likelihood $P(D \mid \Theta, G)$, i.e, the form of the prior is carefully selected to be such that the posterior distribution obeys the form of the prior distribution. For Bayesian networks, the solution is to make the prior distribution to be a product of distributions

$$P(\Theta \mid G, \vec{\alpha}) = \prod_{i=1}^{n} P(\Theta_i \mid G_i, \vec{\alpha}_{G_i}) = \prod_{i=1}^{n}\prod_{j=1}^{q_i} P(\Theta_{ij} \mid \vec{\alpha}_{ij}), \tag{3.1}$$

where the $\vec{\alpha}$ (actually $\vec{\alpha}_G$) has the same structure as Θ, parametrizing the $P(\Theta_{ij} \mid \vec{\alpha}_{ij})$ as a Dirichlet distribution $Dir(\vec{\alpha}_{ij})$:

$$P(\Theta_{ij} \mid \vec{\alpha}_{ij}) = \frac{1}{\mathcal{B}(\vec{\alpha}_{ij})} \prod_{k=1}^{r_i} \Theta_{ijk}^{\alpha_{ijk}-1}. \tag{3.2}$$

The multinomial beta function $\mathcal{B}$ is a constant that does not depend on the parameters Θ_{ij}. It simply acts as a normalizer guaranteeing that the $P(\Theta_{ij} \mid \vec{\alpha}_{ij})$ is a proper density function that integrates to unity, i.e., $\mathcal{B}(\vec{\alpha}_{ij}) = \int \prod_{k=1}^{r_i} \theta_{ijk}^{\alpha_{ijk}-1} d\theta_{ij}$. Due to its form, this prior distribution is easy to combine with the likelihood (Equation 2.3). The resulting posterior probability distribution $P(\Theta \mid G, D)$ is also a product of Dirichlet distributions since $P(\Theta_{ij} \mid D, \vec{\alpha}_{ij}) \sim Dir(\vec{\alpha}_{ij} + \vec{N}_{ij})$.

We still face the problem of specifying the hyperparameter vectors $\vec{\alpha}_{ij}$ that define the distribution of Θ_{ij}. Bayesians are quick to argue that this problem is indeed a virtue that allows us to input background knowledge into the learning system. However, to automate the learning of the parameters from the data, the usual choice is to initially give each instantiation of the Θ_{ij} vector equal probability. This can be accomplished by setting all $\alpha_{ijk} = 1.0$. While the uniform distribution is a convenient choice, the topic of selecting the correct non-informative prior is a favourite pastime of practitioners of different schools of Bayesian theory [3, 6, 5].

For a Bayesian, the posterior distribution is the end result of statistical enquiry about the parameters. From this distribution one can extract the most probable parameter values, expected parameter values, credible intervals, the variance of parameters, etc. Instead of picking certain parameters, the Bayesian answer to the question of data generating parameters is to assign a probability (density) to each choice of parameters.

3.1.3 Parameters for prediction

Much of the appeal of Bayesian networks stems from the ability to infer aspects of the cases which we have not encountered before. Therefore, the parameter learning may aim to restore this ability using the observed data sample. However, picking the maximum likelihood parameters based on small data sample often leads to a poor generalization ability. As an extreme example, if the data matrix D has just one row, and in it the value for variable X_1 is k and the value of X_{G_1} is j, augmenting the network with maximum likelihood parameters yields a Bayesian network that gives zero probability to all the data vectors in which values for X_1 and X_{G_1} do not equal k and j, respectively. Giving zero probabilities to some data vectors after seeing just one data vector is clearly not desirable. This gullible, overfitting behaviour of maximum likelihood estimates makes the approach suboptimal for predictive purposes.

In Bayesian setting, using the posterior distribution to select the most probable parameters alleviates the overfitting problem, but the truly Bayesian approach to prediction would be, instead of selecting particular parameters, to weight predictions given by different parameters by their probability:

$$\begin{aligned}
P(X \mid D, \vec{\alpha}, G) &= \int P(X, \theta \mid D, \vec{\alpha}, G) d\theta \qquad (3.3)\\
&= \int P(X \mid \theta, G) P(\theta \mid D, \vec{\alpha}, G) d\theta\\
&= \int \left[\prod_{i=1}^{n} P(X_i \mid X_{G_i}, \theta_{ij_i}, G_i) \right] P(\theta \mid D, \vec{\alpha}, G) d\theta,
\end{aligned}$$

where $j_i = X_{G_i}$. This *model averaging* calls for integrating over a complicated sets of parameters. Using the parameter independence, the posterior $P(\theta \mid D, \vec{\alpha}, G)$ can be expressed as a product, and we can move the integral in Equation (3.3) inside the product:

$$\begin{aligned} P(X \mid D, \vec{\alpha}, G) &= \prod_{i=1}^{n} \int P(X_i \mid X_{G_i}, \theta_{ij_i}, G_i) P(\theta_{ij_i} \mid D, \vec{\alpha}_{ij_i}, G_i) d\theta_{ij_i} \\ &= \prod_{i=1}^{n} \int \theta_{ij_ik_i} P(\theta_{ij_i} \mid D, \vec{\alpha}_{ij_i}, G_i) d\theta_{ij_i} \\ &= \prod_{i=1}^{n} \tilde{\theta}_{ij_ik_i}, \end{aligned} \tag{3.4}$$

where $\tilde{\theta}_{ij_ik_i}$ is the (a posteriori) expected value of the variable $\Theta_{ij_ik_i}$. Since, a posteriori, each Θ_{ij} is Dirichlet distributed with a hyperparameter vector $\vec{\alpha}_{ij} + \vec{N}_{ij}$, the expected values can be obtained[2] by setting

$$\tilde{\theta}_{ijk} = \frac{\alpha_{ijk} + N_{ijk}}{\sum_{k'=1}^{r_i} \alpha_{ijk'} + N_{ijk'}}. \tag{3.5}$$

Joining the Equations (3.4) and (3.5) leads to a simple method to implement the Bayesian predictive distribution by setting the parameters to their expected values, i.e,

$$P(X \mid D, \vec{\alpha}, G) = P(X \mid \tilde{\theta}(D, \vec{\alpha}, G)) = \prod_{i=1}^{n} \frac{\alpha_{ij_ik_i} + N_{ij_ik_i}}{\sum_{k'=1}^{r_i} \alpha_{ij_ik'} + N_{ij_ik'}}. \tag{3.6}$$

Figure 3.1 features an example of this Bayesian method of learning the parameters.

A predictive parameterization based on sNML

In the research paper V, we propose a non-Bayesian alternative to learning Bayesian network parameters that are good for prediction. The idea is to use the so called sequential normalized maximum likelihood (sNML) parameters that lead to the equation

$$\theta_{ijk} = \frac{e(N_{ijk})(N_{ijk} + 1)}{\sum_{k'=1}^{r_i} e(N_{ijk'})(N_{ijk'} + 1)}, \tag{3.7}$$

where $e(N) = (\frac{N+1}{N})^N$; $(e(0) = 1)$. An example of this method of learning the parameters is shown in Figure 3.1. We will discuss the theory behind the sequential NML later in Chapter 4.

[2] This is a well known property of Dirichlet distributions [25].

	N_{ij1}	N_{ij2}	N_{ij3}
$\vec{N}_{ij}$:	3	7	0

	Θ_{ij1}	Θ_{ij2}	Θ_{ij3}		Θ_{ij1}	Θ_{ij2}	Θ_{ij3}
ML	$\frac{3}{10}$	$\frac{7}{10}$	$\frac{0}{10}$		0.300	0.700	0.000
Bayes	$\frac{4}{13}$	$\frac{8}{13}$	$\frac{1}{13}$	=	0.308	0.615	0.077
sNML	$\frac{210827008}{686047501}$	$\frac{452984832}{686047501}$	$\frac{22235661}{686047501}$		0.307	0.660	0.032

Figure 3.1: Learning the parameters in three different ways for the counts $\vec{N}_{ij} = (3, 7, 0)$. In the Bayesian case, the hyperparameters were set by $\alpha_{ijk} = 1.0$.

3.2 Learning the structure of Bayesian networks

When learning the parameters for a Bayesian network, we assume that the data has been generated from a Bayesian network and that we know the structure of that network. In case we do not know the structure, that too should be learned from the data. However, this task is rather complicated in practice.

First of all, it might be the case (and usually is) that the data has not been generated from any Bayesian network. On the other hand, by suitably setting the parameters of a complete Bayesian network (i.e., any Bayesian network with the maximum number ($\frac{n(n-1)}{2}$) of arcs), it is possible to present any distribution[3], so there is never a way to tell for certain that the data did not come from a complete Bayesian network. However, a complete network structure gives little insight to the domain of interest, and from the probabilistic inference point of view, it amounts to listing the probabilities for all the possible combinations of variables (see Figure 2.4).

If we assume that the data was generated from an unknown Bayesian network, we can pose the question about the structure of the Bayesian network that generated the data sample. We might also ask a more specific question about both the structure and the parameters of the Bayesian network that generated the data [32, 31, 40].

[3] The factorization of the likelihood function given by a complete network corresponds to the chain rule of the probability theory which always holds.

3.2.1 A problem with using maximum likelihood

The classical maximum likelihood principle cannot be used for structure learning. In order to use it, we should be able to find the Bayesian network structure that gives the highest probability to our data sample. However, the quest is nonsensical since the structure alone does not determine any probability for the data, but for that we need both the structure and the parameters.

The question about the structure and the parameters which together give the data sample the highest probability also yields disappointing results since the complete network can always be parametrized so that no simpler structure with any parameters can beat it. The principle of *Occam's Razor* calls for selecting the simplest model among (otherwise) equally good models. In practice, the maximal likelihood for the data can often be achieved with a slightly sparser structure than the complete network. However, the simplest of these structures is still usually far too complex to give good insight about the structure of the domain. The quest for parsimony is often realized by explicitly penalizing the the model for its "complexity". These penalized maximum likelihood models are discussed further in section 3.2.4.

3.2.2 On the nature of network structure

If the network structure alone cannot determine the probability distribution, what is the meaning of the structure? From a purely formal point of view, the network structure constraints the kind of joint probability distributions that can be presented with any parametrization of it. This gives raise to an equivalence relation among the structures: the network structures are considered (distribution) equivalent if the sets of distributions obtainable by their parametrizations are the same. This equivalence can also be be characterized by the properties of the network structure: network structures are equal, if they have the same skeleton and the same set of V-structures [79]. We say that skeletons of networks are the same if after replacing the directed arcs with undirected ones, the networks have the same undirected edges. By V-structure in network structure G we mean triplets of variables (A, B, C) such that there are arcs from A to C, and B to C, but there are no arcs between A and B (neither from A to B, nor from B to A). Each network structure has a set of these triplets, and if for two different networks the sets are the same, we say that they have the same V-structures.

The distributional constraints induced by a network structure can be shown to control the conditional independence relations among the variables in all the distributions that can be obtained by parametrizing the structure [46]. Therefore, a network structure is often seen as presenting a certain set of independence statements. The reason for the characterization being phrased in terms of independence rather than dependence is that a carefully tailored parametrization can produce a distribution in which variables are independent of each other even if there is an arc between them. Therefore, the existence of the arc does not necessarily guarantee the dependence[4].

On the other hand, the missing arc between two variables A and B in network structure G comes with the guarantee that the variables can be made conditionally independent by some conditioning set in all distributions obtainable by parametrizing the structure G. This guarantee may be expressed by saying that the important thing in a Bayesian network structure is the missing arcs, not the arcs that are present [58]. The statement emphasizes the role of independence in producing compact knowledge representation and efficient reasoning. However, it is cognitively hard to focus on things missing from the picture rather than those things present.

From a causal knowledge representation point of view, the structure specifies the variables that force the values of other variables, barring unspecified exceptions that may occur. Unlike statistical dependence, causality is a non-symmetric relation, so the equivalence of network structures described above does not apply to causal Bayesian networks.

The “independence interpretation” of the Bayesian network structure has yielded many algorithms that utilize conditional independence tests for learning the structure [74, 11]. While conceptually well aligned with the “independence interpretation”, these algorithms are subject to intricacies of hypothesis testing, such as selecting significance levels and correcting for multiple hypotheses testing. Furthermore, these methods do not offer a framework to conduct both parameter and structure learning.

Following the focus in research papers I-V, we will concentrate on the so called score-based learning, where the problem can be divided into defining a measure of goodness (often called the score) for the networks, and then using some search algorithm for finding a network structure with an optimal score.

The score based approach is not totally separate from the independence test approach [17], and it is possible to construct hybrid algorithms [20].

[4] However, the existence of the arc from A to B in any network structure G implies that A and B are conditionally dependent with all conditioning sets in almost all distributions obtained by parametrizations of G no matter what the conditioning variable set is.

3.2.3 Bayesian structure learning

The standard Bayesian answer to the structure learning task is to calculate the posterior probability distribution $P(G \mid D)$ of the candidate networks. In practice, obtaining the whole distribution is not feasible due to the huge number of possible network structures even in a case of relatively few variables. The number $b(n)$ of Bayesian networks with n variables can be calculated by recursive formula [64]

$$b(n) = \begin{cases} 1 & \text{if } n = 0, \\ \sum_{k=1}^{n} (-1)^{k+1} \binom{n}{k} 2^{k(n-k)} b(n-k) & \text{if } n > 0. \end{cases}$$

Since any undirected graph can be directed to at least one acyclic graph, we notice that the number of Bayesian networks is larger than $2^{\frac{n(n-1)}{2}}$, which shows that the number grows faster than exponentially. The super exponentiality can be observed in lengths of the figures in Table 3.1 where the number of different Bayesian network structures have been listed for up to 20 variables. Pruning away the equivalent network structures does not help much since the number of equivalence classes is about 20% of the number of all the network structures [26].

Table 3.1: Number of Bayesian network structures as a function of n.

n	number of Bayesian network structures with n nodes
0	1
1	1
2	3
3	25
4	543
5	29281
6	3781503
7	1138779265
8	783702329343
9	1213442454842881
10	4175098976430598143
11	31603459396418917607425
12	521939651343829405020504063
13	18676600744432035186664816926721
14	1439428141044398334941790719839535103
15	237725265553410354992180218286376719253505
16	83756670773733320287699303047996412235223138303
17	62707921196923889899446452602494921906963551482675201
18	99421195322159515895228914592354524516555026878588305014783
19	332771901227107591736177573311261125883583076258421902583546773505
20	2344880451051088988152559855229099188899081192234291298795803236068491263

The pursuit of calculating the probability of a Bayesian network structure would usually proceed by using the Bayes' theorem

$$
\begin{aligned}
P(G \mid D) &= \frac{P(D \mid G)P(G)}{P(D)} \\
&= \frac{P(D \mid G)P(G)}{\sum_{G'} P(D \mid G')P(G')}.
\end{aligned}
\tag{3.8}
$$

The normalizing denominator $P(D)$ does not depend on the structure G, so provided that the numerator can be calculated, it is possible to compare the relative probabilities of Bayesian network structures without calculating the denominator. This makes it possible to search for the most probable network structure. However, even if we can find the most probable structure, the inability to calculate the normalizing constant leaves us no direct way to assess its actual probability. In the case of many variables, the probability of the most probable network may be very small. In Bayesian parlance this translates to saying that we are almost sure the network structure with highest probability is not the structure of the Bayesian network that generated the data.

A Bayesian may be quick to point out that the whole task of selecting a network structure is ill-defined and that the objective should be the probability distribution that correctly quantifies the uncertainty about different network structures. Another way out would be to concentrate on some higher level properties of Bayesian networks, for example, whether the data generating Bayesian network had a particular arc or not. Probabilities of such binary features are within meaningful range [42][5].

The numerator of the Equation (3.8) contains the prior probability $P(G)$ of the network and the so called marginal likelihood $P(D \mid G)$. For learning Bayesian networks from the data, the prior $P(G)$ is often assumed to be uniform, i.e., same for all the different network structures, so when comparing probabilities of different networks, the priors cancel out.[6]

Having conveniently dealt with structure priors, the only thing left to compute is the marginal likelihood $P(D \mid G)$, the very entity that rendered maximum likelihood approach impotent since the structure alone does not define probability for the data. However, a Bayesian can stumble over this

[5] However, due to the limited autonomy of single arcs in Bayesian networks, these probabilities may be problematic to interpret. Furthermore, in general cases these probabilities cannot be easily computed.

[6] There is, however, another school of researchers that tend to set the prior according to the complexity of the network giving higher prior probability to the network structures with less arcs. The exact rationale of this procedure is unknown to the author and in research papers we have always used uniform priors.

block by "integrating parameters out":

$$P(D \mid G) = \int P(D \mid \theta, G) P(\theta \mid G) d\theta.$$

This equation contains the prior probability of the parameters $P(\theta \mid G)$, which is a taboo for frequentists, who consider the parameters to be non-random properties of the world, thus talking about the probability of parameters is not meaningful. Many Bayesians share the same ontological commitments, but since they use the probabilities for describing uncertainty about the world, they feel comfortable with probabilities of parameters, values of which they do not know. (The same argument goes for the probability of the structure, too.)

Using the simplifying assumptions of parameter independence and the conjugacy, which were already made for the Bayesian parameter learning (Equation 3.1), the marginal likelihood $P(D \mid G, \vec{\alpha})$ can be expressed in closed form [8, 32]:

$$P(D \mid G, \vec{\alpha}) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\sum_{k=1}^{r_i} \alpha_{ijk})}{\Gamma(\sum_{k=1}^{r_i} \alpha_{ijk} + N_{ijk})} \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})}, \tag{3.9}$$

where the gamma function $\Gamma(z) = \int_0^\infty x^{z-1} e^{-x} dx$ is a continuous generalization of the factorial with a property $\prod_{t=0}^{T-1}(\alpha + t) = \frac{\Gamma(\alpha+T)}{\Gamma(\alpha)}$. Despite its intricate look, the Equation (3.9) can be simply derived by using the chain rule to express the probability of the data as a chain of predictive distributions (Equation 3.6). This leads to the equation

$$P(D \mid G, \vec{\alpha}) = \prod_{t=1}^{N} P(d^t \mid D^{t-1}, \vec{\alpha}) = \prod_{t=1}^{N} \prod_{i=1}^{n} \frac{\alpha_{ij_i^t k_i^t} + N_{ij_i^t k_i^t}^{t-1}}{\sum_{k'=1}^{r_i} \alpha_{ij_i^t k'} + N_{ij_i^t k'}^{t-1}}, \tag{3.10}$$

where the D^{t-1} denotes the first $t-1$ rows of the data matrix D, and the turbulent, if not unruly, $N_{ij_i^t k_i^t}^{t-1}$ may be deciphered with information that the superscript $t-1$ marks the fact that the counts are calculated from D^{t-1}, and that k_i^t and j_i^t denote the value and the parent configuration of the i^{th} variable in t^{th} data row d^t. The result follows by regrouping the terms of the Equation (3.10) to form products that can be expressed as ratios of gamma-functions.

The interpretation of a Bayesian network structure as a set of independence assumptions leads "naturally" to the requirement that data should not help discriminate between equal network structures. This requirement has severe implications to the form of prior distributions for the Bayesian network parameters [32]. If we further require that all the possible data vectors are equally likely a priori, it can be shown that the prior distribution for the parameters has to be such that all the Dirichlet parameters α_{ijk} of the Equation (3.2) are of the form

$$\alpha_{ijk} = \frac{\alpha}{q_i r_i},$$

where the α is a single positive real number called the equivalent sample size. With this selection of priors, the $P(D \mid G, \alpha)$ is called Bayesian Dirichlet equivalence uniform score (BDeu) [8, 32].

The question of specifying the prior has now been reduced to specifying a single positive real number α. Heckerman et al. [32] suggest a method based on the survey method of Winkler [81] for assessing the value of α, but the procedure requires user to answer a complicated hypothetical question which is probably hard to answer very accurately. Furthermore, giving an accurate answer to the question would probably require information about the domain in which case the idea of pursuing a non-informative prior is not tenable, which authors themselves point out too.

Unfortunately, as shown in the research paper III of this dissertation, the posterior probability distribution of the network structures is very sensitive to the choice of the α parameter. This observation is one of the key motivations for the research papers IV and V. Priors in BDeu also sometimes suggest spurious dependencies even if the data does not support them [75].

Historically, the early Bayesian scores did not assume likelihood equivalence. One of the popular choices for parameter priors was to set all α_{ijk} to 1.0, which yields a uniform distribution for the Θ_{ij} [15]. Unfortunately, setting all parameters α_{ijk} to a constant value does not save us from the sensitivity problem – the most probable network structure is still very sensitive to the selection of this constant [7].

3.2.4 Search and decomposability

Since learning the optimal Bayesian network structure is NP-hard for all popular goodness criteria [12], in practice we have to resort to heuristic search strategies. One of the simplest and most often used search strategies is the stochastic greedy search [13]. The search starts with an initial network and finds its goodness (often called score). The search then proceeds by trying out small modifications to the initial network. If the best of these modified networks is better than the initial network, it is selected as a

[7] This is an unpublished result of ours.

new "initial" network to be enhanced again by similar small modifications. If the small modifications do not seem to produce any better networks, the system selects a new initial network and starts the procedure all over again. The initial networks can be specific networks like the empty network, the full network, or one of the best networks with at most one parent, which can be found in reasonable time ($O(n^2)$) [14]. One may also use a modified version of some previously found good network as a new initial network.

Common small modifications include adding, deleting and reversing arcs. These modifications can be implemented efficiently if the scoring criterion has a property of *decomposability*, i.e., the score of the network can be presented as a sum of local scores that measure the goodnesses of individual variables and their parents. More specifically, the score $S(G, D)$ is called decomposable if it can be expressed as a sum of terms, one term per variable, where the term for i^{th} variable depends only on the data columns D_i and D_{G_i}:

$$S(G, D) = \sum_{i=1}^{n} s(D_i, D_{G_i}).$$

If the score is decomposable, after a modification to the network structure, we need to recalculate terms for only those variables whose parents have changed. In practice, this speeds up the search significantly.

By taking the logarithm of the marginal likelihood $P(D \mid G)$ (Equation 3.9), we see that the Bayesian score is decomposable. The research paper I presents an online tool B-course[8] that uses BDeu-score for learning Bayesian network structures.

Some decomposable scores can be found among the so called penalized maximum likelihood scores, where

$$S(G, D) = \log P(D \mid \hat{\theta}(D, G)) - \mathit{penalty}.$$

In the popular Akaike Information Criterion (AIC) [1], the *penalty* equals the dimension $\Delta = \prod_{i=1}^{n} \prod_{j=1}^{q_i} (r_i - 1)$, i.e., the number of free parameters in the model. In another popular scoring criterion, Bayesian Information Criterion (BIC) [69], the penalty term is $\frac{\Delta}{2} \log N$. Both AIC and BIC have been derived by asymptotics, so that the selected network structure should have desirable properties when the number N of rows in the data matrix D grows large (goes to infinity). Unlike Bayesian framework, these structure learning criteria do not suggest any particular way of learning the parameters for the selected structure.

[8] `http://b-course.cs.helsinki.fi/`

Not all the scoring criteria are decomposable. For example, the normalized maximum likelihood (NML) score defines the penalty as “flexibility” of the network structure, $penalty = \log \sum_{D'} P(D' \mid \hat{\theta}(D', G))$, which yields a non-decomposable score. In the research paper IV, we present a decomposable scoring criterion that is based on the NML score. We will discuss this score more in Chapter 4.

Decomposability of the score also makes it possible to find the optimal network structure for up to about 30 variables. The research paper II details an algorithm for this “exact” structure learning. A demonstration and implementation of the algorithm is also freely available.[9]

[9] `http://b-course.cs.helsinki.fi/bene/`

Chapter 4

Minimum description length

In the previous chapter, we briefly mentioned a sensitivity problem when using the BDeu score for learning Bayesian networks. This problem is studied in research paper III. A solution to the problem is presented in the research paper IV. The solution is based on the method of normalized maximum likelihood (NML) [71, 62] that is one of the central constructs in the minimum description length (MDL) approach to statistical inquiry [61, 27]. The NML philosophy is also used in the research paper V for learning parameters for a Bayesian network.

While it is not possible to give a comprehensive introduction to a broad and fundamental subject like MDL in just one chapter, the deployment of NML in research papers calls for some exposition of the topic. For a more rigorous, yet convenient introduction, the reader is advised to study the book *The Minimum Description Length* [27] by Peter Grünwald.

In the following we will shed light to the philosophical background of MDL and NML. We will also highlight differences and similarities between the Bayesian approach and the NML approach for learning Bayesian networks. The key results of the research papers IV and V will be briefly described, but for a more thorough discussion of the results, the reader is advised to consult the papers themselves.

4.1 MDL and rhetorics

In its naive form, Bayesianism assumes models to represent the underlying, unknown reality that produces observations. Based on these observations, we can then infer what the reality is like. The classical statistics also shares this commitment to the idea of reality producing data. The MDL principle is different: it does not assume a reality but uses models as means to describe regularities in data. Different models can describe different kinds of regularities, which raises the question about the criterion for evaluating the models.

The MDL principle gains its objectivity by fixing a criterion for a good description of a data: the model M gives a good description of the data D, if the description is short compared to descriptions (given by M or other models) of other data sets of the same size. It is generally entertained that in order to give a short description of D (i.e., to compress D), the model M has to be able to separate the regularities (information) in D from the features not describable by M (noise). This idea is in harmony with the original motivation of probabilistic modelling of being able to state rules with unspecified exceptions.

The optimality requirement of MDL is not asymptotic. This makes it different from classical frequentism and Bayesian statistics. In frequentism, the whole concept of probability is based on the asymptotic behaviour of the relative frequency. While the Bayesian machinery is capable of dealing with arbitrarily small data sets, a central part of its justification lies in the update rule guaranteeing that the believes of rational agents converge in the limit.

4.2 Normalized maximum likelihood

The method of normalized maximum likelihood is a concrete way to implement the model selection in the spirit of the MDL principle. To set it in context of the previous chapter, the NML principle is described here as a criterion for selecting Bayesian network structures. However, there are currently no known algorithms for computing this criterion efficiently for general Bayesian networks, the fact that has motivated the concept of *factorized NML* in the research paper IV.

The MDL principle aims at concise description of the data, and the notion of short description can be translated to the notion of high probability. Giving frequently occurring items (i.e, items with high probability) short descriptions produces short total descriptions. It is no accident that frequent words in natural language tend to be short [85], or that in Morse code, common letters are coded with short sequences of dots and dashes.

The question is what do we mean by the structure G giving a relatively high probability to the data D. As we noticed when discussing maximum likelihood, the structure itself does not define the probability of the data, but a set of probability distributions that can be obtained by different parametrizations of the structure. Bayesian interpretation allows us to use probability theory for defining a marginal likelihood of data D, which is the average probability assigned to D by distributions expressible by the structure G. Thus, assuming equal priors for the structures, the Bayesian answer is to select the network structure whose distributions on average give data D the highest probability.

In NML, the structure G is not characterized by how its distributions behave on average on data D, but how much higher probability can any of the distributions of G give to D compared to what distributions of G can give to other data sets D'. This formulation avoids the need of prior, and it gives rise to a so called NML distribution

$$P_{\text{NML}}(D \mid G) = \frac{P(D \mid \hat{\theta}(D,G))}{\sum_{D'} P(D' \mid \hat{\theta}(D',G))}, \tag{4.1}$$

where $\hat{\theta}(D,G)$ denotes the maximum likelihood parameters, i.e., the parametrization of G that gives the data D the maximal probability.

The Bayesian marginal likelihood $P(D \mid G)$ is a distribution that "mimics" the behaviour of the whole set of distributions hosted by G by being a weighted average of those distributions. The P_{NML} has an analogous characterization with respect to the distributions hosted by G. It can be shown that for all data sets D of size N, the P_{NML} distribution always assigns D a constant $C(N)$ times lower probability than the maximum likelihood distribution in G does. This constant relationship to the maximum likelihoods obtainable within the distributions of G makes it a unique distribution with a minimax *regret* property

$$P_{\text{NML}}(G) = \operatorname*{argmin}_{Q} \max_{D'} \log \frac{P(D' \mid \hat{\theta}(D',G))}{Q(D')},$$

where Q may be any distribution. $P_{\text{NML}}(G)$ itself is not usually among the distributions expressible by G.

The MDL principle calls for selecting the structure G, the NML distribution of which gives the highest probability to the data D. While the numerator of the Equation (4.1) defines G's ability to fit the data D, the denominator defines the complexity of the G as its ability to fit any given data D' of the same size. This definition of complexity also provides insight into why strong parameter priors may (or even should) influence Bayesian model selection. Priors tame the fitting ability of the model, which lowers the complexity of the model.

Unlike the Bayesian model selection that depends on no other data than the observed one, the NML-denominator explicitly depends on the other data we might have seen, thus it does not adhere to the so called likelihood principle [4].

It may provide some insight to forge the Bayesian criterion into the format of the penalized maximum likelihood

$$\log P(D \mid G) = \log P(D \mid \hat{\theta}(D,G)) - \log \frac{P(D \mid \hat{\theta}(D,G))}{P(D \mid G)}.$$

The Bayesian "penalty" term depends on the data D, and it is large for those network structures in which the maximum likelihood parameters in G assign the data D a much larger probability than distributions in G do on average. It may not be obvious how to interpret this Bayesian penalty as the complexity of the model. It does, however, signal how "peaked" the distribution is at its maximum likelihood point. This peakedness gets it rigorous handling in the concept of Fisher Information [41].

4.3 Factorized NML

While NML is an appealing alternative criterion for learning Bayesian network structures, there is no known algorithm for its efficient exact calculation, and approximate methods using sampling are computationally too demanding to be used in search [66].

Taking the logarithm of the P_{NML} (Equation 4.1) transforms it to the penalized maximum likelihood, but the form of the denominator does not allow an easy factorization that would make the P_{NML} decomposable. This has motivated us to seek more indirect ways to use NML for learning Bayesian network structures.

Insisting on decomposability, the research paper IV proposes a factorized version of NML in which the maximum likelihood of each column of the data matrix is normalized separately

$$\begin{aligned}
P_{\text{fNML}}(D \mid G) &= \prod_{i=1}^{n} P_{\text{NML}}(D_i \mid \hat{\theta}(D_i, D_{G_i}, G)) \qquad (4.2) \\
&= \prod_{i=1}^{n} \frac{P(D_i \mid \hat{\theta}_i(D_i, D_{G_i}, G))}{\sum_{D'_i} P(D'_i \mid \hat{\theta}_i(D'_i, D_{G_i}, G))} \\
&= \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{P(D_i^{G_i=j} \mid \hat{\theta}_{ij}(D_i^{G_i=j}, D_{G_i}, G))}{\sum_{D'_i} P(D'_i \mid \hat{\theta}_{ij}(D_i^{\prime G_i=j}, D_{G_i}, G))} \\
&= \prod_{i=1}^{n} \prod_{j=1}^{q_i} P_{\text{NML}}(D_i^{G_i=j} \mid \hat{\theta}_{ij}(D_i^{G_i=j}, G)).
\end{aligned}$$

The factorized NML criterion is clearly decomposable. Furthermore, it reduces the calculation of the criterion to a product of one dimensional multinomial NML distributions which can be computed efficiently [44].

4.4 Sequential NML

Going for a small regret has also driven the development of NML-based predictive schemes. The sequential NML (sNML) distribution [63, 67] is derived by requiring the predictive distribution to always have a small regret

$$P_{\text{sNML}}(\cdot \mid D, G) = \min_{Q} \max_{d'} \log \frac{P(d' \mid \hat{\theta}(d', D, G))}{Q(d')}.$$

In general, the distribution P_{sNML} is not among the distributions representable by the structure G. Furthermore, restricting Q to those distributions that can be presented with G, i.e., a solution to the problem

$$\min_{Q \in G} \max_{d'} \log \frac{P(d' \mid \hat{\theta}(d', D, G))}{Q(d')},$$

does not necessarily define a unique distribution.

The general idea of sequential NML is used in the research paper V to come up with an sNML based parametrization. The proposed parametrization, *factorized sequential NML* (fsNML), is derived by applying sequential NML to each variable separately

$$P_{\text{fsNML}}(X \mid D, G) = \prod_{i=1}^{n} P_{\text{sNML}}(X_i \mid D_i^{G_i = j_i}, G). \tag{4.3}$$

The proposed solution is analogous to fNML (Equation 4.2), and it is also very easy to compute (see equation 3.7).

Chapter 5

Summary and the background of research papers

In previous chapters we have reviewed the theory of Bayesian networks to the extent needed for understanding the research papers in part II of this dissertation. After a short motivation and the tedious notation for Bayesian networks, the treatise concentrated on learning networks automatically from the data. The main focus was in Bayesian approach since, in addition to being currently the most popular approach, it also provides a framework for both parameter and structure learning.

During the theory overview, we have also provided pointers to the research papers. These pointers usually marked some problematic aspects of the Bayesian approach thus motivating the development detailed in the papers. The proposed solutions to some of these problems were briefly sketched in Chapter 4 where the fNML criterion and the fsNML parametrization was proposed as efficient and objective methods for learning Bayesian networks.

In what follows, I will provide some personal insight to the actual process that lead to each of these papers; something that is usually carefully hidden from the published work. I hope this will make reading the papers more enjoyable.

5.1 Paper I

> P. Myllymäki, T. Silander, H. Tirri, and P. Uronen. B-course: A web-based tool for Bayesian and causal data analysis. *International Journal on Artificial Intelligence Tools*, 11(3):369–387, 2002.

Paper I describes B-course, a web-site that hosts an interactive tutorial about learning Bayesian networks from the data. The paper exists only because many people who used B-course asked for a reference to it, and for a long time there was none. The real contribution is the web-site, `http://b-course.cs.helsinki.fi/`, the paper is just academic clutter. Much of the paper is actually written by Myllymäki and Tirri, while the algorithms behind the tool are written by me, and the user interface by Pekka Uronen who revised and improved my original user interface.

Early on (1997 –) CoSCo research group was interested in data analysis for social sciences. This interest was mostly tunneled via group leader, professor Henry Tirri whose wife, professor Kirsi Tirri, was conducting educational research. Educational researchers needed tools, not only theorems, and there were not many tools for Bayesian data analysis available at that time. One of our first tools was a Naive Bayes classifier BAYDA [45] which, even if not supported for many years now, is still being downloaded regularly. With more than 3000 identified downloaders this tool was a stand-alone Java application that featured a wizard-like interface which at the time was a sharp contrast to the usual data analysis software.

The downside of the BAYDA was that the Java behaved differently on different platforms. The idea of B-course was to provide a server side implementation of the data analysis tool which would be easier to maintain. The B-course, a tool for constructing Bayesian networks from the data, was first introduced at the American Educational Research Association conference, AERA 2000 where it was used in tutorials on Bayesian data analysis by Henry Tirri and me. I cannot help mentioning that, at the department, Henry was severely bashed by tenured professors for letting me to implement the B-course since it had no "academic value". Admittedly, the theory for learning Bayesian network structures had been around for some years, but there were no good implementations for practitioners around. CoSCo had a strong tradition in heuristic search methods and Bayesian data analysis, and a tradition of implementing data analysis methods.

Since year 2000, B-course has been used in many courses and tutorials. The first tutorials were held in American Educational Research Association Conferences (AERA 2000, AERA 2002). After that, Petri Nokelainen has held at least six courses (two in Tampere, two in Helsinki, and two in Tallinn) with B-course. The tool has also frequently been a part of our own courses at the computer science department of the University of Helsinki. B-course has also found its way into several doctoral dissertations [29, 35, 53, 22]. Domestically, in his own doctoral dissertation [54], Petri Nokelainen refers to six other Finnish dissertations on educational research in which B-course has been utilized. The B-course was meant to be a tutorial, but many people have found it convenient and convincing enough to be used as a tool for research. Therefore, B-course appears in several research papers both with us [77, 23, 34, 55] but also by others working in domains we know very little about [37, 36, 65, 10, 84]. The work by others we have just accidentally found since B-course can be used freely without any registration or login.

5.2 Paper II

> T. Silander and P. Myllymäki. A simple approach for finding the globally optimal Bayesian network structure. In R. Dechter and T. Richardson, editors, *Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence (UAI-06)*, pages 445–452. AUAI Press, 2006.

Learning the most probable Bayesian network structure is an NP-hard problem, and that is why heuristic search is usually used to find a good network. A brute force method of evaluating all the possible network structures is feasible only for less than ten variables. It must have been professor Henry Tirri who told me that professor Heikki Mannila had said that using methods developed by academy research fellow Mikko Koivisto, it is possible to find the most probable network structure for up to 25 variables [43]. That was an interesting result, but when I tried to read Mikko's paper about the subject, I failed to understand much of it. Furthermore, it was concentrating on summing over the network structures and not finding the most probable structure. It may well be that it was an easy corollary of sum-product-wizardry to convert that algorithm to the max-product case, but at that time it was too much for me.

However, while struggling to understand Mikko's paper, I came across a very simple dynamic programming idea of finding the most probable network. I implemented the prototype of the program in Python, and even with this simple implementation, it was possible to find the most probable networks for 18 variables or so. The idea was so simple that someone else probably had invented it before, but I was not familiar with the algorithm, so it could not be very well known one. With a more careful implementation, I was able to learn networks for 30 variables.

Petri Myllymäki then persuaded me to write a paper about the algorithm for the UAI 2006 conference, which was a good idea since, to my understanding, it was not well known in that community that it is indeed possible to find the "globally optimal" Bayesian network structure up to 30 variables using common decomposable scoring criteria.

While still writing the paper, Petri found a technical report of Singh and Moore [73] describing the very same algorithm. Their implementation was probably not that optimized so the empirical work was not so impressive, but the algorithm was definitely the same. I wonder if they had found the trivial idea to be too simple to be published in some other way. Later, when reviewing an article for JAIR, I came across another result that contains the very same idea [56], but because of its publication forum, we had been ignorant about it. Anyway, at UAI 2007 professor Kevin Murphy thanked me for publishing this simple algorithm since his group at the University of British Columbia had made an implementation of it in their own research software.

5.3 Paper III

> T. Silander, P. Kontkanen, and P. Myllymäki. On sensitivity of the MAP Bayesian network structure to the equivalent sample size parameter. In R. Parr and L. van der Gaag, editors, *Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence (UAI-07)*, pages 360–367. AUAI Press, 2007.

Already in the UAI 2006 paper, we found that it was important to give extra motivation for presenting a non-scalable, exponential time and space algorithm for learning small Bayesian networks. The truth was that while for us the sheer possibility of learning a little bit bigger networks was motivating enough, for rest of the community the case was probably not equally intriguing. We then ended up arguing that the new algorithm was valuable in studying the properties of the "optimal networks". Previously, due to the heuristic search, we could not know if the network found was actually optimal, so this kind of study was not easy to conduct.

A natural candidate for such a study was the role of the commonly used hyperparameter in the BDeu score. The common practice was to select this parameter value to be 1.0. Steck and Jaakkola had earlier shown analytically that if the parameter approaches zero or infinity, some unwanted phenomena happen [76]. However, most of the people probably neglected these findings since the results were asymptotic.

We then conducted a rather extensive series of experiments and demonstrated that there was a severe sensitivity problem in the BDeu score. A very small change in the hyperparameter α often changed the structure of the most probable model, and this happened in all of the 20 data sets we used.

We decided to publish this result at the UAI 2007 conference since it was probably a new significant result for the community. Even before the UAI 2007, Harald Steck had heard about our result (from Petri Myllymäki I guess) and he wanted to read the paper beforehand. After that he asked for the data sets we had used to conduct his own studies on the subject. His results were published in the UAI 2008 [75]. After Steck's talk Pedro Domingos made a statement "This means that the BDeu score is broken!", a statement that Harald Steck tried to soften by saying that "one has to be careful when using it." Anyway, we had naturally drawn similar conclusions, and we were working on the solution.

5.4 Paper IV

> T. Silander, T. Roos, P. Kontkanen, and P. Myllymäki. Factorized normalized maximum likelihood criterion for learning Bayesian network structures. *Proceedings of the 4th European Workshop on Probabilistic Graphical Models (PGM-08)*, pages 257–264, Hirtshals, Denmark, 2008.

The background of the Paper IV presents another line of research practiced in CoSCo group, namely the minimum description length (MDL). With a long time collaboration with professors Jorma Rissanen and Peter Grünwald, the group has developed a special expertise in implementing information theoretic model selection criteria. The idea for this paper was conceived by my CoSCo colleague, Petri Kontkanen and professor Petri Myllymäki at the bar. Petri Kontkanen, who had developed an algorithm for efficient calculation of normalized maximum likelihood (NML) for a single multinomial variable, had noticed that this criterion can be calculated for a single conditional distribution as well. Petri Myllymäki made an obvious comment that a Bayesian network is nothing more than a collection of these conditional distributions. In a traditional CoSCo Friday session the new finding was discussed, and we realized that this new, factorized NML (fNML) was a decomposable score that can be efficiently calculated by our dynamic programming algorithm.

I then implemented the score in our software, and Teemu Roos, our group's expert on everything MDL, set the score to the context of recently discovered sequential NML models. We then wrote a paper on the subject for Jorma Rissanen's Festschrift [50], but since that work is unlikely to reach many people in graphical models community, we decided to rewrite it for the PGM conference for which we also ran more experiments.

After many iterations and rewriting, the story now goes that we "propose a decomposable form of normalized maximum likelihood criterion since it allows us to use our exact learning algorithm and heuristic search". The truth is that we would like to calculate the actual NML criterion, but we cannot, and that this rationalization of "proposing" fNML came only after we had noticed that this is something we are able to compute.

Anyway, the results for structure learning are very promising. The conference paper got a good reception, and we were asked to submit a journal version of it to the International Journal of Approximate Reasoning, which we gladly did.[1]

5.5 Paper V

> T. Silander, T. Roos, and P. Myllymäki. Locally minimax optimal predictive modeling with Bayesian networks. In D. van Dyk and M. Welling, editors, *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS-09), Volume 5 of JMLR: W&CP 5*, pages 504–511, Clearwater Beach, Florida, USA, 2009.

But the story is not finished yet. We had found a way out of the parameter sensitivity problem in the Bayesian structure learning, but we knew no better way to assign the parameters to the networks but the Bayesian way. Intellectually this was very dissatisfying. After bashing Bayesian solution to make our fNML solution look good, how could we justify using Bayesian solution to the parameter learning. The Bayesian way of setting parameters is practically a consequence of the Bayesian structure learning criterion. Having refuted the latter, it would be very hard to defend the former.

The answer came to me when I discussed fNML with Teemu Roos. Since he had worked with Jorma Rissanen on the sequential NML (sNML), he saw the similarity between sequential and factorized versions of the NML, probably aiming at the more general theory about normalizing maximum likelihood in parts. I had earlier entertained ideas about different local regularizations in the Bayesian network structure learning (some of that work is yet to be published), and sNML as a new predictive scheme just found a correct slot in my mind.

We decided to submit a paper on this topic to AISTATS 2009. Process of writing the paper produced nice negative results about some obvious alternatives to the proposed factorized sequential NML (fsNML) parametrization, which further justified our proposal. Empirical results turned out to be encouraging, and we were even able to prove some performance

[1] The paper is currently under review.

bounds. The paper was accepted for the conference at which we also learnt that our work on NML based methods had sparked a reading group on NML at the University of British Columbia.

5.6 Storyline and a summary of contributions

To conclude the Part I of this dissertation, here is the summary of the papers in a format that tries to reveal a storyline and highlight the main contributions of the papers:

Paper I We introduce B-course, an online tool for learning Bayesian networks from the data. B-course is consequently used in several courses, research papers, and doctoral dissertations. The tool uses the BDeu score and a heuristic stochastic search algorithm, so there are no guarantees that it actually finds the most probable network structure.

I implemented the website and the algorithms. Most of the paper is written by professors Tirri and Myllymäki. The current version of the interface is strongly revised by Pekka Uronen.

Paper II We develop a dynamic programming algorithm that is guaranteed to find the optimal network for less than 30 variables. This makes it possible to compare different scoring criteria and to study properties of the optimal networks.

The work for this paper is mostly done by me.

Paper III We use our new exact structure learning algorithm introduced in paper II to find out that the BDeu model selection criterion we (among others) have been using has a severe sensitivity problem.

The work for this paper is mostly done by me.

Paper IV We propose an efficient objective scoring criterion, the factorized NML, that is free of the sensitivity problem discussed in paper III. The new criterion is based on the MDL principle. It is consistent, and the empirical tests demonstrate its good behaviour.

The idea came from Petri Kontkanen and Petri Myllymäki. I implemented the exact structure learning for it and did the experiments. The paper is mostly written by me. Teemu Roos had a significant role in reformulating the NML part (he is an expert on that).

Paper V Paper IV calls for developing a new objective parametrization rule for the networks. Again, we tap into the MDL theory and develop a factorized sequential NML parametrization scheme. The new scheme is easy to implement, and it features good predictive performance.

The paper is mostly written by me. Teemu Roos had a significant role in formulating the sNML part (he is an expert on that too).

References

[1] H. Akaike. Information theory and an extension of the maximum likelihood principle. In B. Petrox and F. Caski, editors, *Proceedings of the Second International Symposium on Information Theory*, pages 267–281, Budapest, 1973. Akademiai Kiado.

[2] S. Andersen, K. Olesen, F. Jensen, and F. Jensen. Hugin – a shell for building belief universes for expert systems. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1080–1085, Detroit, Michigan, August 1989. Morgan Kaufmann Publishers, San Mateo, CA.

[3] J. Berger and J. Bernardo. On the development of reference priors. In J. Bernardo, J. Berger, A. Dawid, and A. Smith, editors, *Bayesian Statistics 4*, pages 35–60. Oxford University Press, 1992.

[4] J. Berger and R. L. Wolpert. *The Likelihood Principle.* Institute of Mathematical Sciences, Hayward, California, 1988.

[5] J. Bernardo. Noninformative priors do not exist. *J. Statist. Planning and Inference*, 65:159–189, 1997.

[6] J. Bernardo and A. Smith. *Bayesian theory.* John Wiley, 1994.

[7] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics).* Springer, August 2006.

[8] W. Buntine. Theory refinement on Bayesian networks. In B. D'Ambrosio, P. Smets, and P. Bonissone, editors, *Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence*, pages 52–60. Morgan Kaufmann Publishers, 1991.

[9] E. Castillo, J. Gutiérrez, and A. Hadi. *Expert Systems and Probabilistic Network Models.* Monographs in Computer Science. Springer-Verlag, New York, NY, 1997.

[10] Y. Chen, Q. Q., and Q. Chen. Learning dependency model for amp-activated protein kinase regulation. In *Knowledge Science, Engineering and Management*, pages 221–229. Springer Berlin / Heidelberg, 2007.

[11] J. Cheng, R. Greiner, J. Kelly, D. Bell, and W. Liu. Learning Bayesian networks from data: An information-theory based approach. *Artificial Intelligence J.*, 137(1-2):43–90, 2002.

[12] D. Chickering. Learning Bayesian networks is NP-Complete. In D. Fisher and H. Lenz, editors, *Learning from Data: Artificial Intelligence and Statistics V*, pages 121–130. Springer-Verlag, 1996.

[13] D. Chickering. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3:507–554, 2002.

[14] C. Chow and C. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions in Information Theory*, 14(3):462–467, May 1968.

[15] G. Cooper and E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309–347, 1992.

[16] R. Cowell, P. Dawid, S. Lauritzen, and D. Spiegelhalter. *Probabilistic Networks and Expert Systems*. Springer, New York, NY, 1999.

[17] R. G. Cowell. Conditions under which conditional independence and scoring methods lead to identical selection of Bayesian network models. In J. S. Breese and D. Koller, editors, *Proceedings of the Seventeenth Annual Conference on Uncertainty in Artificial Intelligence (UAI–01)*, pages 91–97. Morgan Kaufmann, 2001.

[18] A. Darwiche. Recursive conditioning. *Artificial Intelligence*, 126(1-2):5–41, February 2001.

[19] A. Dawid. Conditional independence in statistical theory. *Journal of the Royal Statistical Society B*, 41:1–31, 1979.

[20] L. M. de Campos. A scoring function for learning Bayesian networks based on mutual information and conditional independence tests. *Journal of Machine Learning Research*, 7:2149–2187, 2006.

[21] R. Dechter. Bucket elimination: A unifying framework for probabilistic inference. In E. Horvits and F. Jensen, editors, *"Proceedings of the Twelfth Annual Conference on Uncertainty in Artificial Intelligence (UAI–96)*, pages 211–219, Portland, Oregon, August 1996. Morgan Kaufmann Publishers, San Francisco, CA.

[22] K. Deforche. *Modeling HIV resistance evolution under drug selective pressure.* PhD thesis, Katholieke Universiteit Leuven, 2008.

[23] K. Deforche, T. Silander, R. Camacho, Z. . Grossman, M. A. Soares, K. V. Laethem, R. Kantor, Y. Moreau, and A.-M. Vandamme. Analysis of HIV-1 pol sequences using Bayesian networks: implications for drug resistance. *Bioinformatics*, 22(24):2975–2979, 2006.

[24] K. Fuchi, R. Kowalski, K. Furukawa, K. Ueda, K. Kahn, T. Chikayama, and E. Tick. Launching the new era. *Commun. ACM*, 36(3):49–100, 1993.

[25] A. Gelman, J. Carlin, H. Stern, and D. Rubin. *Bayesian Data Analysis.* Chapman & Hall, 1995.

[26] S. B. Gillispie. Enumerating Markov equivalence classes of acyclic digraph models. In *Proc. of the Conf. on Uncertainty in Artificial Intelligence*, pages 171–177. Morgan Kaufmann, 2001.

[27] P. Grünwald. *The Minimum Description Length Principle.* MIT Press, 2007.

[28] T. M. Haavelmo. The statistical implications of a system of simultaneous equations. *Econometrica*, 11:1–11, 1943.

[29] A. Hartemink. *Principled Computational Methods for the Validation and Discovery of Genetic Regulatory Networks.* PhD thesis, Massachusets Institute of Technology, 2001.

[30] D. Heckerman. Probabilistic interpretation for MYCIN's certainty factors. In L. Kanal and J. Lemmer, editors, *Uncertainty in Artificial Intelligence 1*, pages 167–196, Amsterdam, 1986. Elsevier Science Publishers B.V. (North-Holland).

[31] D. Heckerman. A tutorial on learning with Bayesian networks. Technical Report MSR-TR-95-06, Microsoft Research, Advanced Technology Division, One Microsoft Way, Redmond, WA 98052, 1996.

[32] D. Heckerman, D. Geiger, and D. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20(3):197–243, September 1995.

[33] C. Howson and P. Urbach. *Scientific Reasoning: The Bayesian Approach.* Open Court, Chicago, 1993.

[34] M. Jaeger, J. D. Nielsen, and T. Silander. Learning probabilistic decision graphs. *Int. J. Approx. Reasoning*, 42(1-2):84–100, 2006.

[35] A. Jakulin. *Machine Learning Based on Attribute Interactions.* PhD thesis, University of Ljubljana, Faculty of Computer and Information Science, June 2005.

[36] S. Jaroszewicz and D. A. Simovici. Interestingness of frequent itemsets using Bayesian networks as background knowledge. In *KDD '04: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 178–186, New York, NY, USA, 2004. ACM.

[37] E. Jarvis, V. Smith, K. Wada, M. Rivas, M. McElroy, T. Smulders, P. Carninci, Y. Hayashizaki, F. Dietrich, X. Wu, P. McConnell, J. Yu, P. Wang, A. Hartemink, and S. Lin. A framework for integrating the songbird brain. *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, 188(11–12):961–980, December 2002.

[38] E. T. Jaynes and G. L. Bretthorst. *Probability Theory as Logic.* Cambridge University Press, 2003.

[39] F. V. Jensen and T. D. Nielsen. *Bayesian Networks and Decision Graphs.* Information Science and Statistics. Springer Verlag, 2nd edition, 2007.

[40] M. Jordan. *Learning in graphical models.* The MIT Press, The Netherlands, 1998.

[41] R. Kass and P. Voss. *Geometrical Foundations of Asymptotic Inference.* Wiley Interscience, 1997.

[42] M. Koivisto. Advances in exact Bayesian structure discovery in Bayesian networks. In *Proceedings of the 22nd Annual Conference on Uncertainty in Artificial Intelligence (UAI-06)*, pages 241–248, Arlington, Virginia, 2006. AUAI Press.

[43] M. Koivisto and K. Sood. Exact Bayesian structure discovery in Bayesian networks. *Journal of Machine Learning Research*, 5:549–573, May 2004.

[44] P. Kontkanen and P. Myllymäki. A linear-time algorithm for computing the multinomial stochastic complexity. *Information Processing Letters*, 103(6):227–233, 2007.

[45] P. Kontkanen, P. Myllymäki, T. Silander, and H. Tirri. BAYDA: Software for Bayesian classification and feature selection. In R. Agrawal, P. Stolorz, and G. Piatetsky-Shapiro, editors, *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98)*, pages 254–258. AAAI Press, Menlo Park, 1998.

[46] S. Lauritzen. *Graphical Models.* Oxford University Press, 1996.

[47] S. Lauritzen and D. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *J. Royal Stat. Soc., Ser. B*, 50(2):157–224, 1988. Reprinted as pp. 415–448 in [70].

[48] J. McCarthy. Programs with common sense. In *Proceedings of the Teddington Conference on the Mechanization of Thought Processes*, pages 75–91. Her Majesty's Stationary Office, London, 1959.

[49] J. McCarthy. First order theories of individual concepts and propositions. *Machine Intelligence*, 9:129–148, 1979.

[50] P. Myllymäki, T. Roos, T. Silander, P. Kontkanen, and H. Tirri. Factorized NML models. In P. Grünwald, P. Myllymäki, I. Tabus, M. Weinberger, and P. Yu, editors, *Festschrift in Honor of Jorma Rissanen*, pages 189– 204. TICSP Series #38, Tampere International Center for Signal Processing, 2008.

[51] R. Neapolitan. *Probabilistic Reasoning in Expert Systems.* John Wiley & Sons, New York, NY, 1990.

[52] R. E. Neapolitan. *Learning Bayesian Networks.* Prentice Hall, 2003.

[53] W. Ng'ang'a. *Word Sense Disambiguation of Swahili: Extending Swahili Language Technology with Machine Learning.* PhD thesis, University of Helsinki, 2005.

[54] P. Nokelainen. *Modeling of Professional Growth and Learning: Bayesian approach.* PhD thesis, Tampere University, 2008.

[55] P. Nokelainen, T. Silander, P. Ruohotie, and H. Tirri. Investigating the number of non-linear and multi-modal relationships between observed variables measuring growth-oriented atmosphere. *Quality and Quantity*, 41(6):869–890, December 2007.

[56] S. Ott and S. Miyano. Finding optimal gene networks using biological constraints. *Genome Informatics*, 14:124–133, 2003.

[57] J. Pearl. Bayesian networks: A model of self-activated memory for evidential reasoning. In *Proceedings of the 7th Conference of the Cognitive Science Society, University of California, Irvine*, pages 329–334, August 1985.

[58] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann Publishers, San Mateo, CA, 1988.

[59] J. Pearl. *Causality: Models, Reasoning and Inference.* Cambridge University Press, 2000.

[60] J. Pearl and A. Paz. Graphoids: Graph-based logic for reasoning about relevance relations. In B. Du Boulay, D. Hogg, and L. Steels, editors, *Advances in Artificial Intelligence-II*, pages 357–363. North-Holland, Amsterdam, 1987.

[61] J. Rissanen. *Stochastic Complexity in Statistical Inquiry.* World Scientific Publishing Company, New Jersey, 1989.

[62] J. Rissanen. Fisher information and stochastic complexity. *IEEE Transactions on Information Theory*, 42(1):40–47, January 1996.

[63] J. Rissanen and T. Roos. Conditional NML universal models. In *Proceedings of the Information Theory and Applications Workshop (ITA-07)*, pages 337–341, San Diego, CA, January–February 2007. IEEE Press.

[64] R. Robinson. Counting unlabeled asyclic graphs. In C. Little, editor, *Combinatorial Mathematics*, number 622 in Lecture Notes in Mathematics, pages 28–43. Springer-Verlag, 1977.

[65] A. Rodin, T. H. Mosley, A. G. Clark, C. F. Sing, and E. Boerwinkle. Mining genetic epidemiology data with Bayesian networks application to apoe gene variation and plasma lipid levels. *Journal of Computational Biology*, 12(1):1–11, February 2005.

[66] T. Roos. Monte carlo estimation of minimax regret with an application to MDL model selection. In *Proceedings of the 2008 IEEE Information Theory Workshop (IEEE–08)*, pages 284–288, May 2008.

[67] T. Roos and J. Rissanen. On sequentially normalized maximum likelihood models. In *Workshop on Information Theoretic Methods in Science and Engineering (WITMSE-08)*, Tampere, Finland, 2008.

[68] S. Russel and P. Norvig. *Artificial Intelligence: A Modern Approach.* Prentice Hall, New Jersey, 1995.

[69] G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6:461–464, 1978.

[70] G. Shafer and J. Pearl, editors. *Readings in Uncertain Reasoning.* Morgan Kaufmann Publishers, San Mateo, CA, 1990.

[71] Y. Shtarkov. Universal sequential coding of single messages. *Problems of Information Transmission*, 23:3–17, 1987.

[72] H. A. Simon. Causal ordering and identifiability. In W. Hood and T. Koopmans, editors, *Studies in Econometric Method*, pages 49–74. Wiley, 1953.

[73] A. Singh and A. Moore. Finding optimal Bayesian networks by dynamic programming. Technical report, Carnegie Mellon University, June 2005.

[74] P. Spirtes, C. Glymour, and R. Scheines, editors. *Causation, Prediction and Search.* Springer-Verlag, 1993.

[75] H. Steck. Learning the Bayesian network structure: Dirichlet prior vs data. In D. A. McAllester and P. Myllymäki, editors, *Proceedings of the 24th Annual Conference on Uncertainty in Artificial Intelligence (UAI–98)*, pages 511–518. AUAI Press, 2008.

[76] H. Steck and T. S. Jaakkola. On the Dirichlet prior and Bayesian regularization. In *Advances in Neural Information Processing Systems 15*, pages 697–704, Vancouver, Canada, 2002. MIT Press.

[77] R. S. Thomas, D. R. Rank, P. S. G., G. M. Zastrow, K. R. Hayes, K. Pande, E. Glover, T. Silander, M. W. Craven, J. K. Reddy, S. B. Jovanovich, and C. A. Bradfield. Identification of toxicologically predictive gene sets using cDNA microarrays. *Molecular pharamacology*, 60(6):1189–1194, December 2001.

[78] R. Thomason. Logic and artificial intelligence. In E. N. Zalta, editor, *The Stanford Encyclopedia of Philosophy.* Fall 2008.

[79] T. Verma and J. Pearl. Equivalence and synthesis of causal models. In *Proceedings of the Sixth Annual Conference on Uncertainty in Artificial Intelligence (UAI-90)*, pages 255–270, New York, NY, USA, 1991. Elsevier Science Inc.

[80] J. Whittaker. *Graphical Models in Applied Multivariate Statistics.* John Wiley & Sons, 1990.

[81] R. Winkler. The assessment of prior distributions in Bayesian analysis. *American Statistical Association Journal*, 62:776–800, 1967.

[82] S. Wright. Correlation and causation. *Journal of Agricultural Research*, 20:557–585, 1921.

[83] S. Wright. The method of path coefficients. *Annals of Mathematical Statistics*, 5:161–215, 1934.

[84] J. Zhao, J. Luan, M. F. Baksh, and Q. Tan. Mining gene networks with application to gaw15 problem 1. In *BMC Proceedings, 1 (Suppl 1): S52.* BioMed Central Ltd., 2007.

[85] G. K. Zipf. *The psycho-biology of language: an introduction to dynamic philology.* The MIT Press, Cambridge, MA, 1935.

Part II

Research papers included in the dissertation