Workshop proceedings

Probabilistic Graphical Models

PGM
2018
Prague

Organized by:
Institute of Information Theory and Automation,
Czech Academy of Sciences, Prague

Prague

September 11, 2018

Published by:
ÚTIA AV ČR, v.v.i.,
Institute of Information Theory and Automation, Czech Academy of Sciences
Pod Vodárenskou věží 4, 182 08 Praha 8, Czech Republic

in Prague — September 2018

Organized by:
Institute of Information Theory and Automation, Czech Academy of Sciences, Prague

Credits:
Editors: Václav Kratochvíl, Milan Studený
LaTeXeditor: Václav Kratochvíl
using LaTeX's 'confproc' package, version 0.8 by V. Verfaille

Programme Committee:

Milan Studený - chair, *Czech Academy of Sciences, Czech Republic*
Václav Kratochvíl - co-chair, *Czech Academy of Sciences, Czech Republic*

Alessandro Antonucci, *IDSIA, Switzerland*
Concha Bielza Lozoya, *Universidad Politécnica de Madrid, Spain*
Janneke Bolt, *Utrecht University, Netherlands*
Cory Butz, *University of Regina, Canada*
Andrés Cano, *University of Granada, Spain*
Arthur Choi, *University of California, Los Angeles, USA*
Barry Cobb, *Missouri State University, USA*
Giorgio Corani, *IDSIA (Istituto Dalle Molle di Studi sull'Intelligenza Artificiale, Switzerland*
Fabio Cozman, *University of Sao Paulo, Brazil*
James Cussens, *University of York, UK*
Cassio De Campos, *Queen's University Belfast, UK*
Luis M. de Campos, *University of Granada, Spain*
Nicola Di Mauro, *Universita di Bari, Italy*
Francisco Javier Díez, *UNED, Spain*
Marek Druzdzel, *University of Pittsburgh, USA & Bialystok University of Technology, Poland*
Robin Evans, *University of Oxford, UK*
Ad Feelders, *Utrecht University, Netherlands*
José A. Gámez, *University of Castilla-La Mancha, Spain*
Manuel Gómez Olmedo, *University of Granada, Spain*
Anna Gottard, *University of Florence, Italy*
Arjen Hommersom, *Open University of the Netherlands, Netherlands*
Antti Hyttinen, *University of Helsinki, Finland*
Mohammad Ali Javidian, *University of South Carolina, USA*
Frank Jensen, *HUGIN EXPERT, Denmark*
Radim Jiroušek, *University of Economics, Czech Republic*
Mikko Koivisto, *University of Helsinki, Finland*
Johan Kwisthout, *Radboud University, Netherlands*
Helge Langseth, *Norwegian University of Science and Technology, Norway*
Pedro Larranaga, *University of Madrid, Spain*
Philippe Leray, *LINA/DUKe - Nantes University, France*
Jose A. Lozano, *The University of the Basque Country, Spain*
Peter Lucas, *Radboud University, Netherlands*
Manuel Luque, *UNED, Spain*
Marloes Maathuis, *ETH Zurich, Switzerland*
Anders L Madsen, *HUGIN EXPERT, Denmark*
Brandon Malone, *NEC Laboratories Europe, Germany*
Radu Marinescu, *IBM Research, Ireland*
Andrés Masegosa, *University of Granada, Spain*
Maria Sofia Massa, *University of Oxford, UK*
Denis Mauá, *University of Sao Paulo, Brazil*
Serafín Moral, *University of Granada, Spain*
Thomas Dyhre Nielsen, *Aalborg University, Denmark*
Ann Nicholson, *Monash University, Australia*

Foreword

These proceedings contain papers to be presented in the form of workshop contributions within the 9th International Conference of Probabilistic Graphical Models (PGM 2018). The papers are interpreted as working (versions of the) papers: they describe some work in progress.

We wish all the participants in the conference PGM 2018 a pleasant stay in Prague.

In Prague, September 11, 2018 *Milan Studený and Václav Kratochvíl*

Contents

Teaching Bayesian networks with OpenMarkov

Francisco Javier Díez FJDIEZ@DIA.UNED.ES
Iago París IAGOPARIS@DIA.UNED.ES
Jorge Pérez-Martín JPEREZMARTIN@DIA.UNED.ES
Manuel Arias MARIAS@DIA.UNED.ES
Dept. Artificial Intelligence. Universidad Nacional de Educación a Distancia (UNED). Madrid. Spain

Abstract

OpenMarkov is an open-source software tool for probabilistic graphical models. It has been developed especially for medicine, but it has also been used for building applications in other fields, in a total of more than 30 countries. In this paper we explain how to use it as a pedagogical tool to teach the main concepts of Bayesian networks, such as conditional dependence and independence, d-separation, Markov blankets, explaining away, etc., and some inference algorithms: logic sampling, likelihood weighting, and arc reversal. The facilities for learning Bayesian networks interactively can be used to illustrate step by step the performance of the two basic algorithms: search-and-score and PC.

Keywords: OpenMarkov; Bayesian networks; d-separation; inference; learning Bayesian networks.

1. Introduction

Bayesian networks (BNs) (Pearl, 1988) and influence diagrams (Howard and Matheson, 1984) are two types of probabilistic graphical models (PGMs) widely used in artificial intelligence. Unfortunately, the theory that supports them is complex. Our computer science students, in spite of their relatively strong mathematical background, find it hard to intuitively grasp some of the fundamental concepts, such as conditional independence and d-separation. Additionally, we have been teaching PGMs to health professionals, most of them medical doctors, for more than two decades, and although we avoid the more complex aspects (for instance, we do not speak of d-separation and only teach them the variable elimination algorithm), some of the basic notions important for them, such as conditional independence, are difficult to convey. In this paper we show how OpenMarkov, an open-source tool with an advanced graphical user interface (GUI) has allowed us to make more intuitive some concepts that we found very difficult to explain before we had it.

The rest of this paper is structured as follows: Section 2 introduces the background (notation, definitions, and an overview of OpenMarkov), Section 3, the core of the paper, explains how to teach BNs, Section 4 presents a brief discussion, and Section 5 contains the conclusion.

2. Background

2.1 Basic notation and definitions

In this paper we represent variables with capital letters (X) and their values with lower-case letters (x). A bold upper-case letter ($\mathbf{X}$) denotes a set of variables and a bold lower-case letter ($\mathbf{x}$) denotes a configuration of them, i.e., the assignment of a value to each variable in $\mathbf{X}$. In this paper we assume

that each variable has a finite set of values, called states. When a variable X is boolean, we denote by $+x$ the state "true", "present", or "positive", and by $\neg x$ the state "false", "absent", or "negative".

Two variables X and Y are (a priori) *independent* when

$$\forall x,\, \forall y,\;\; P(x,y) = P(x) \cdot P(y)\,. \tag{1}$$

When $P(y) \neq 0$ for a particular value of Y, this implies that

$$\forall x,\;\; P(x \mid y) = P(x)\,, \tag{2}$$

i.e., knowing the value taken by Y does not alter the probability of X.

We say that two variables X and Y are *conditionally independent* given a set of variables $\mathbf{Z}$, and denote it as $I_P(X, Y \mid \mathbf{Z})$, when

$$\forall x,\, \forall y,\, \forall \mathbf{z},\;\; P(x,y \mid \mathbf{z}) = P(x \mid \mathbf{z}) \cdot P(y \mid \mathbf{z})\,. \tag{3}$$

In a directed graph, when there is a link $X \to Y$, we say that X is a parent of Y and Y is a child of X. The set of parents of a node X is denoted by $Pa(X)$, and $pa(X)$ represents a configuration of them. When there is a directed path from X to Y, we say that X is an ancestor of Y and Y is a descendant of X.

2.2 Bayesian networks

A PGM consists of a graph and a probability distribution, $P(\mathbf{v})$, such that each node in the graph represents a variable in $\mathbf{V}$; for this reason we often speak indifferently of nodes and variables. The relation between the graph and the distribution depends on the type of PGM: a BN, a Markov network, an influence diagram, and so forth.

In the case of a BN, the graph is directed and acyclic, and its relation with the probability distribution is given by the following properties; we can take any one of them as the definition of a BN and then prove that the other two derive from it (Pearl, 1988; Neapolitan, 1990; Koller and Friedman, 2009):

1. **Factorization of the probability:** The joint probability is the product of the probability of each node conditioned on its parents, i.e.,

$$P(\mathbf{v}) = \prod_{X \in \mathbf{V}} P(x \mid pa(X))\,.$$

2. **Markov property.** Each node is independent of its non-descendants given its parents, i.e., if $\mathbf{Y}$ is a set of nodes such that none of them is a descendant of X, then

$$P(x \mid pa(X), \mathbf{y}) = P(x \mid pa(X))\,.$$

3. **d-separation.** If two nodes X and Y are d-separated in the graph given a set of nodes $\mathbf{Z}$, which we denote by $I_G(X, Y \mid \mathbf{Z})$, then they are probabilistically independent given $\mathbf{Z}$:

$$\forall X,\, \forall Y,\, \forall \mathbf{Z},\;\; I_G(X, Y \mid \mathbf{Z}) \Longrightarrow I_P(X, Y \mid \mathbf{Z})\,.$$

   Two nodes are *d-separated* when there is no active path connecting them. A path is *active* if every node W between X and Y satisfies this property:

   (a) if the arrows that connect W with its two neighbors converge in it, then W or at least one of its descendants is in $\mathbf{Z}$;

   (b) else, W is not in $\mathbf{Z}$.

2.3 OpenMarkov

OpenMarkov (see `http://www.openmarkov.org`) is a software tool for PGMs developed at the National University for Distance Education (UNED) in Madrid, Spain. It consists of around 115,000 lines of Java code (excluding comments and blanks), structured in 44 maven sub-projects and stored in a git repository at Bitbucket. The first versions were distributed under the European Union Public Licence (EUPL), version 1.1, while version 0.3 of OpenMarkov and the next ones will be distributed under the GNU public license, version 3 (GPLv3).

It offers support for editing and evaluating several types of PGMs, such as BNs (Pearl, 1988), influence diagrams (Howard and Matheson, 1984), Markov influence diagrams (Díez et al., 2017), and decision analysis networks (Díez et al., 2018). Its native format for encoding the networks is ProbModelXML (Arias et al., 2012).

It has been designed mainly for medicine; with OpenMarkov and its predecessor, Elvira (Elvira Consortium, 2002), our group has built models for more than 10 real-world medical problems, each involving dozens of variables. Some groups have used it to build PGMs in other fields, such as planning and robotics (Oliehoek et al., 2017). To our knowledge, it has been used at universities, research institutions, and large companies in more than 30 countries.

2.4 Evidence propagation in BNs with OpenMarkov

In a diagnostic problem, the assignment of a value to a variable as a consequence of an observation is called a *finding*. The set of findings is called *evidence*. The *propagation* of evidence consists in computing the posterior probability of some variables given the evidence.

In OpenMarkov chance variables are drawn as rounded rectangles and colored in cream, as shown in Figure 1. When a finding is introduced (usually by double-clicking on the value of the variable), OpenMarkov propagates it and shows the posterior probability of every state of every variable by means of a horizontal bar. It is possible to have several sets of findings, each called an *evidence case*, and display several bars for every state.

3. Teaching Bayesian networks

3.1 Basic concepts of probability and BNs

3.1.1 Correlation and independence

Even though the concepts of probabilistic dependence (correlation) and independence are mathematically very simple (cf. Eqs. 1 and 3), many students have difficulties to understand them intuitively, especially in the case of conditional independence. In our teaching, we use the network in Figure 1, which has a clear causal interpretation: all the variables are boolean, and for each link $X \rightarrow Y$ the finding $+x$, i.e., the presence of X, increases the probability of $+y$, except in the case of vaccination, $+v$, which decreases the probability of the second disease, $+d_2$.

We begin by explaining that in this model the two viruses, V_A and V_B, are supposed to be causally and probabilistically *independent* because there is no link between them and they have no common ancestor. We can check it by introducing a finding for virus A and observing that the probability of V_B does not change (cf. Eq. 2); for example, $P(+v_B|+v_A) = P(+v_B|\neg v_A) = P(+v_B) = 0.01$, as shown in Figure 1. In contrast, we can see that the variables V_A and D_1 are *correlated* by introducing evidence about the one and observing that probability of the other

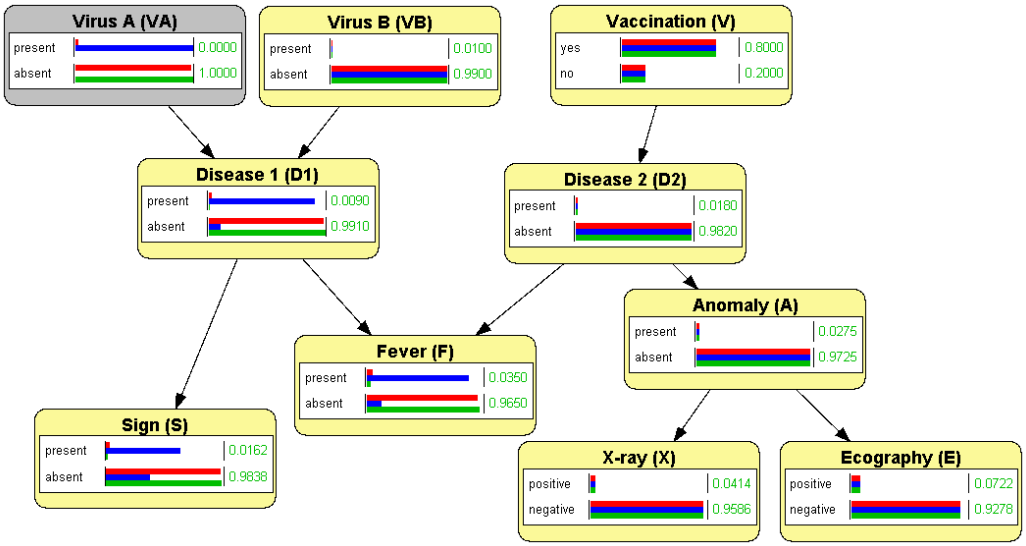


Figure 1: A Bayesian network for the differential diagnosis of two hypothetical diseases. In this model V_A and V_B are a priori independent. We can check it by introducing evidence about V_A and observing that the probability of V_B, represented by horizontal colored bars, does not change. The same holds for the 5 variables at the right of F. In contrast, the 4 descendants of V_A do depend on the evidence for this variable.

changes; for example, in Figure 1 we observe that $P(+d_1|+v_A) = 0.9009 > P(+d_1) = 0.0268 > P(+d_1 \mid \neg v_A) = 0.009$.

In order to illustrate the concept of conditional independence, we first show that S and F are correlated by introducing evidence on S and seeing that the probability of F changes. However, if we first introduce evidence about D_1, which plays the role of the conditioning variable, the evidence about S does not alter the probability of F, as we can observe in Figure 2, which shows that F and S, in spite of being correlated a priori, are conditionally independent given D_1. Our students easily understand the correlation between fever and the sign is due to a common cause, and when we know with certainty whether this cause is present or absent, the correlation disappears. OpenMarkov confirms that our intuitive understanding of causation leads to the numerical results we expected.

3.1.2 D-SEPARATION

In Section 2.2 we have introduced the definition of d-separation based on the concept of active paths. If we leave the students just with this mathematical definition, they are absolutely unable to understand the rationale behind it—we would also be! In particular, it is difficult to understand why if the arrows that connect a node W with its two neighbors converge in W then this node or some of its descendants must be in $\mathbf{Z}$ to make this path active, while if the arrows do not converge in W then it is the opposite, i.e., W cannot be in $\mathbf{Z}$.

In order to solve this puzzle, we explain that in this context $\mathbf{Z}$ represents a set of observed variables. We then analyze how the definition of d-separation applies when the path containing just one link; in this case there is no node between X and Y, so the path is active, by definition.

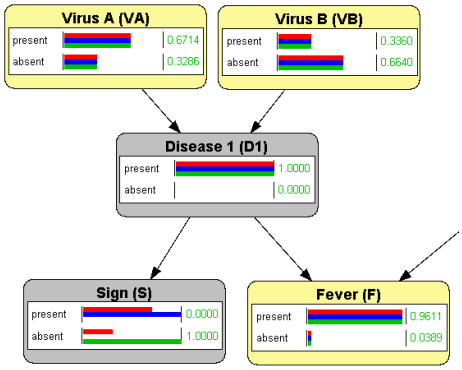


Figure 2: In this network V_A and V_B are a priori independent. We can check it by introducing evidence about V_A and observing that the probability of V_B does not change. The same holds for the 5 variables at the left of F. In contrast, the descendants of V_A do depend on the evidence for this variable.

We then consider a path consisting of two links, sometimes called a *trail* (Koller and Friedman, 2009), which can be of three types: divergent, convergent, and sequential. A trail is divergent when both links depart from the node in the middle; for example, $S \leftarrow D_1 \rightarrow F$. When there is no evidence, i.e., when $\mathbf{Z} = \varnothing$, the path is active and therefore $\neg I_G(S, F \mid \varnothing)$, which allows S and F to be correlated;[1] we can check that they are in fact correlated by introducing evidence for one of them. In contrast, if we have a finding for D_1, then $\mathbf{Z} = \{D_1\}$, and $I_G(S, F \mid \{D_1\})$ implies $I_P(S, F \mid \{D_1\})$, as we have seen in the previous subsection (cf. Fig. 2). The behavior of a sequential trail is similar; for example, the path $V_A \rightarrow D_1 \rightarrow S$ is active when there is no finding for D_1, because $\neg I_G(V_A, S \mid \varnothing)$ of d-separation, but any finding about D_1 blocks this path: $I_G(V_A, S \mid \{D_1\})$. So the causal interpretation of this path agrees with the properties of dependence and independence that derive from the definition of d-separation.

Let us consider now a convergent trail, such as $V_A \rightarrow D_1 \leftarrow V_B$. We can check that it is inactive when $\mathbf{Z} = \varnothing$ by introducing evidence for V_A and observing that the probability of V_B, as we did in Figure 1, which is quite intuitive, because there is no common cause for these variables. In contrast, if we introduce first evidence about D_1, then this trail becomes active, $\neg I_G(V_A, V_B \mid \{D_1\})$; we can observe it by introducing evidence about V_A and observing that the probability of V_B changes. In particular, $P(+v_B \mid +d_1, +v_A) < P(+v_B \mid +d_1) < P(+v_B \mid +d_1, \neg v_A)$. This also agrees with the causal interpretation of the BN, because when a patient has the first disease, we suspect that the cause is virus A or virus B; if additional evidence (for example, the result of a test) leads us to discarding virus A, we then suspect that the cause of the disease is virus B, but if the presence of A is confirmed, of our suspicion of B decreases. Put another way, V_A and V_B are a priori independent, but the finding $+d_1$ introduces a negative correlation between them. This phenomenon, called *explaining away* (Pearl, 1988), is the most typical case of intercausal reasoning; in particular, it is a property of the noisy-OR model (Pearl, 1988; Díez and Druzdzel, 2006). (In this network we have a noisy OR at D_1 and another one at F.)

1. We say "allows S and F to be correlated" instead of "are correlated" because the separation in the graph implies probabilistic independence, but the reverse is not true.

We can also observe that the convergent trail $V_A \rightarrow D_1 \leftarrow V_B$ is not only activated by D_1 itself, but also by any of its descendants. In fact, the explaining-away phenomenon also occurs for $+s$ and $+f$, because either of these findings makes us suspect the presence of at least one of the viruses, thus establishing a negative correlation between V_A and V_B. In contrast, D_1 can block the divergent trail $S \leftarrow D_1 \rightarrow F$, but the ancestors of D_1 cannot: $\neg I_G(S, F \mid \{V_A, V_B\})$. We can check by first introducing evidence about V_A and/or V_B and then observing that S and F are still correlated. This also agrees with our intuitive notion of causality because in this model both viruses increase the probability of $+d_1$ but none of them confirms definitely its presence; so $+s$ further increases the probability of $+d_1$ and, consequently, that of $+f$.

3.1.3 Markov property and Markov blankets

As we saw in Section 2.2, the Markov property means that every node is conditionally independent of its non-descendants given its parents. We can use again the network in Figure 1 to check that this property holds for every node; in particular, a node having no parents is a priori independent of its non-descendants.

Similarly, we can use our example network to illustrate the concept of Markov blanket, which denotes a set of nodes that surround a node making it conditionally independent of the other variables in the network (Pearl, 1988). Intuitively, the set of parents and children of a node D_1 form a Markov blanket for it. However we can see that this is not the case: if we introduce evidence for V_A, V_B, S, and F, we can see that D_1 is not yet separated from all the other nodes in the network; in fact, every node in $\{V, D_2, A, X, E\}$ is correlated with D_1 because F has activated the trail $D_1 \rightarrow F \leftarrow D_2$. Therefore, the Markov blanket of a node must include not only its parents and children, but also its children's parents.

3.1.4 The back-door path in causal models

One of the most difficult tasks in observational studies is to infer causal relations. Typically, when there is an unobserved common cause U of two variables X and Y, one might erroneously conclude that X is a cause of Y or vice versa; in this context, U is called a *confounder*. A randomized controlled trial that manipulates X and observes Y can avoid the confusion, but in many cases it is not possible to conduct that experiment due to temporal, budgetary, or ethical constraints, and even when possible, the analysis of the underlying causal relations is not always trivial. For this reason Pearl (2000) proposed using BNs as a tool for the analysis. The following example illustrates how intuition can be wrong in an apparently simple case.

Let us assume that an epidemiologist has observed, by means of randomized clinical trials, that X is a cause of Y, and Y is a cause of Z. In order to gain more insight about the underlying causal mechanisms, he re-examines his database, thinking that $I_P(X, Z \mid Y)$ will imply that the influence of X on Z is mediated only by Y, while $\neg I_P(X, Z \mid Y)$ will prove that X is able to cause Z by means of an alternative causal mechanism. This reasoning seems very intuitive, but is wrong.

We can explain it using the causal network in Figure 3, in which the only causal path from X to Z passes through Y; U is an observed cause of both Y and Z. We can check that $P(+z|+y, +x) > P(+z|+y, \neg x)$, i.e., $\neg I_P(X, Z \mid Y)$. However, this correlation between X and Z (given Y) is not due to a causal mechanism other than that mediated by Y. The reason for the confusion is that conditioning on Y unwillingly opens a *back-door path* (Pearl, 2000) responsible for a spurious—i.e.,

non-causal—correlation between X and Z even when conditioning on Y (rather, due to the conditioning on Y, because previously the back-door path was not active).

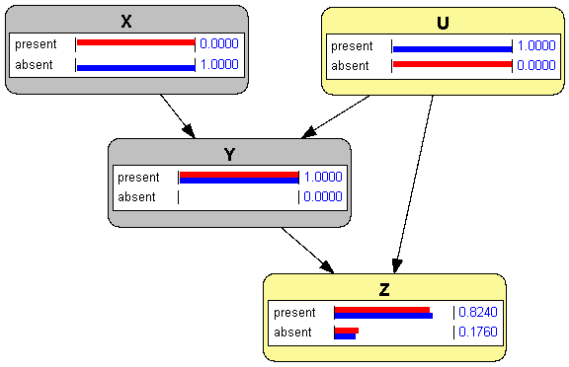


Figure 3: Illustration of the back-door path. In this network the only causal path from X to Z passes through Y. However, when conditioning on Y we see that $\neg I_P(X, Z \mid Y)$, which might lead to the wrong conclusion that there is another causal mechanism from X to Z.

3.2 Inference algorithms

Inference algorithms can be used to propagate evidence in BNs, i.e., to compute the posterior probability of some variables. In addition to teaching our students the most common algorithms, namely variable elimination and clustering, we also explain other algorithms that may interesting for different reasons, discussed in Section 4. Using again the BN in Figure 1, we show here how to compute the probability of the first disease for a patient with fever and the sign, who was not vaccinated, i.e., $P(+d_1|+f, +s, \neg v)$, applying two stochastic algorithms and one exact method.

3.2.1 Stochastic algorithms

OpenMarkov currently implements two stochastic algorithms: logic sampling (Henrion, 1988) and likelihood weighting (Fung and Chang, 1990). Both of them begin by sampling a value for each node without parents, in accordance with its prior distribution, and then proceed in topological order (i.e., downwards) sampling each other node in accordance with the probability distribution for the configuration of its parents. This way, each iteration of the algorithm obtains a sample, which is a configuration of all the nodes. OpenMarkov is able to store these configurations in a spreadsheet and compute some statistics, including the posterior probability of each variable.

Figure 4 shows the result of evaluating the network in Figure 1 with the evidence $\{+f, +s, \neg v\}$. In logic sampling (left side), the variables have been sampled in the topological order $\{V_A, V_B, D_1, V, D_2, F, S, A, X, E\}$. The 10,000 configurations obtained are stored in the "Samples" sheet, with a sample per row and a variable per column; those compatible with the evidence are colored in green and those incompatible in red. The tab "General stats" shows that only 37 samples are compatible, a clear indication of the inefficiency of this algorithm.

For each variable, the spreadsheet shows the number of samples in which each state has appeared. The sum for all the states of a variable is the total number of samples, obviously. It also

	A	B	C
1	**Bayesian network**	two-diseases.pgmx	
2	**Approximate algorithm**	logic sampling	
3	**Total number of samples**	10,000	
4	**Computing time (ms)**	24.25	
5	**Computing time per sample (ms)**	0.0024	
6	**Non-null samples**	37	
7	**Accumulated weight**	37.00	
8	**Exact algorithm**	clustering	
9			
10	**Variable**	**States**	
11			
12	Virus A (VA)	absent	present
13	number of ocurrences	9,773	227
14	approximate probability	0.2778	0.7222
15	exact probability	0.3482	0.6518
16			
17	Virus B (VB)	absent	present
18	number of ocurrences	9,898	102
19	approximate probability	0.7222	0.2778
20	exact probability	0.6738	0.3262
21			
22	Disease 1 (D1)	absent	present
23	number of ocurrences	9,714	286
24	approximate probability	0.0278	0.9722
25	exact probability	0.0293	0.9707
26			
27	Vaccination (V)	no	yes
28	number of ocurrences	2,063	7,937
29	approximate probability	1.0000	0.0000
30	exact probability	1.0000	0.0000
31			
32	Disease 2 (D2)	absent	present
33	number of ocurrences	9,815	185
34	approximate probability	0.9444	0.0556
35	exact probability	0.9254	0.0746
36			
37	Fever (F)	absent	present
38	number of ocurrences	9,464	536
39	approximate probability	0.0000	1.0000
40	exact probability	0.0000	1.0000
41			
42	Sign (S)	absent	present
43	number of ocurrences	9,728	272

Samples | General stats

	A	B	C
1	**Bayesian network**	two-diseases.pgmx	
2	**Approximate algorithm**	likelihood weighting	
3	**Total number of samples**	10,000	
4	**Computing time (ms)**	26.46	
5	**Computing time per sample (ms)**	0.0026	
6	**Non-null samples**	10,000	
7	**Accumulated weight**	188.15	
8	**Exact algorithm**	clustering	
9			
10	**Variable**	**States**	
11			
12	Virus A (VA)	absent	present
13	number of ocurrences	9,807	193
14	approximate probability	0.3475	0.6525
15	exact probability	0.3482	0.6518
16			
17	Virus B (VB)	absent	present
18	number of ocurrences	9,902	98
19	approximate probability	0.6778	0.3222
20	exact probability	0.6738	0.3262
21			
22	Disease 1 (D1)	absent	present
23	number of ocurrences	9,729	271
24	approximate probability	0.0290	0.9710
25	exact probability	0.0293	0.9707
26			
27	Disease 2 (D2)	absent	present
28	number of ocurrences	9,494	506
29	approximate probability	0.9018	0.0982
30	exact probability	0.9254	0.0746
31			
32	Anomaly (A)	absent	present
33	number of ocurrences	9,408	592
34	approximate probability	0.8843	0.1157
35	exact probability	0.9176	0.0824
36			
37	X-ray (X)	negative	positive
38	number of ocurrences	9,303	697
39	approximate probability	0.8849	0.1151
40	exact probability	0.9157	0.0843
41			
42	Ecography (E)	negative	positive
43	number of ocurrences	9,019	981

Samples | General stats

Figure 4: Output of the stochastic algorithms logic sampling (left) and likelihood weighting (right). The latter only samples the variables that do not make part of the evidence.

shows the posterior probability, which is not proportional to the number of occurrences of the state because many samples have been rejected. In particular, the probability for a state compatible with (i.e., included in) the evidence is 1, provided that there is at least one valid sample.

Figure 4 (right side) shows the output of the likelihood weighting algorithm, which only samples the variables that do not make part of the evidence. The first difference we observe is that now the number of non-null samples is the same as the total number of samples, because all the samples are valid. The weight of each sample is between 0 and 1, as we can see in the "Samples" sheet. As a consequence, the total weight for this network and this evidence is 188.15, much higher than the value of 37 obtained for logic sampling (because that algorithm only obtained 37 valid samples, each with a weight of 1), and this in turn leads to more accurate estimates of the posterior probabilities.

3.2.2 ARC REVERSAL

Arc reversal was initially designed for transforming influence diagrams into decision trees (Howard and Matheson, 1984). Later Olmsted (1983) designed an algorithm that iteratively removes the nodes from the influence diagram, one by one, until only the utility node remains—see also (Shachter, 1986). This basic idea can also be applied to computing the posterior probability of interest X in a BN by eliminating all the nodes that are neither the variable of interest nor evidence variables. A barren node (i.e., one without children) can be deleted directly; a node having children can be made barren by inverting its outgoing links. In the final step, the arcs outgoing from X are inverted and then the conditional probability table for this node contains $P(x \mid \mathbf{e})$, the probability of interest.

Version 0.3 of OpenMarkov's GUI will offer not only the possibility of deleting a node, as in any other tool, but also an option for inverting a link $X \to Y$ when both $P(x|pa(X))$ and $P(y|pa(Y))$ are in the form of probability tables.

As an example, we can apply this method to compute in the GUI $P(+d_1|+f, +s, \neg v)$, the same posterior probability that we estimated with the two stochastic algorithms. First of all, we remove the barren nodes, X and E, which converts A into a barren node, ready to be deleted. In order to eliminate V_A, we invert the link $V_A \to D_1$. Then OpenMarkov adds a link $V_B \to V_A$ (because V_B is a parent of D_1) and computes the new conditional probabilities for V_A and D_1 as follows:

$$P(d_1 \mid v_B) = \sum_{v_A} P(v_A, d_1 \mid v_B) = \sum_{v_A} P(v_A) \cdot P(d_1 \mid v_A, v_B)\,,$$

$$P(v_A \mid v_b, d_1) = P(v_A, d_1 \mid v_B)/P(d_1 \mid v_B)\,.$$

Then V_A is a barren node, which we can deleted. We then invert the link $V_B \to D_1$, which adds no link because none of these nodes has other parents; OpenMarkov computes the new conditional probabilities, $P(d_1)$ and $P(v_B|d_1)$. We then delete V_B. The last node to be eliminated is D_2, which has one child, F. When we ask OpenMarkov to invert the arc $D_2 \to F$, it adds the links $D_1 \to D_2$ and $V \to F$ and computes the new conditional probabilities. After removing D_2 we obtain a BN with three links: $D_1 \to S$, $D_1 \to F$, and $V \to F$. Inverting the first one does not add any new link, but the reversal of $D_1 \to F$ adds the links $S \to F$ and $V \to D_1$. The conditional probability table for D_1 is $P(d_1|f, s, v)$, in which we can observe that $P(+d_1|+f, +s, \neg v) = 0.9707$, the same value that OpenMarkov obtains with variable elimination or clustering.

3.3 Learning Bayesian networks

BNs can be built from human knowledge, from data, or from a combination of both. OpenMarkov implements the two basic algorithms for learning BNs from data: search-and-score (Cooper and Herskovits, 1992) and PC (Spirtes and Glymour, 1991). Other tools offer many more algorithms, but the advantage of OpenMarkov is the possibility of interactive learning: the GUI shows a list of the edit (operations) it is ready to perform, with a motivation for each, so that the user can observe how the algorithm proceeds, step by step, and either accept the next operation proposed by the algorithm, or select another one from the list, or do a different edit at the GUI.

The search-and-score algorithm, also called "hill climbing", departs from a network with a node for each variable in the data, and no link. The possible edits are adding a directed link, or deleting or inverting one of those already present in the network. This process is guided by a metric chosen by the user. Currently OpenMarkov offers six well-known metrics: BD, Bayesian, K2, entropy, AIC, and MDLM. When learning the network, it selects the edits compatible with the restrictions of the

network (for example, a BN cannot have cycles) and ranks them according to their scores. This way, a student can see, for example, that when the network has no link yet, the metric K2 usually assigns different scores to the links $X \rightarrow Y$ and $Y \rightarrow X$, even though the networks resulting represent exactly the same probability distribution, which is an unsatisfactory property of this metric. It is also possible to see how the addition of a link usually changes scores for the addition, removal, or reversal of nearby links.

In contrast the PC algorithm departs from a fully connected undirected graph and removes the links one by one depending on the conditional independencies found in the database. For each link X–Y, OpenMarkov performs a statistical test that returns the p-value for the hypothesis that X and Y are a priori independent; if p is below a certain threshold, α, called the significance level, the link is kept; otherwise, it is removed. It then tests, for each pair of variables, whether they are independent given a third variable, and then given a pair of other variables, and so on. In each of these steps the GUI shows the user a list of the links that might be removed, together with the p and the conditioning variables for it. This way, the user can not only see the removals that the algorithm is considering, but also the motivation for each one. Finally, the algorithm assigns a direction to each link.

The tutorial of OpenMarkov, available at `www.openmarkov.org/docs/tutorial`, explains in detail the options it offers for learning BNs, either automatically or interactively.

4. Discussion

Some networks that required stochastic evaluations in the past can now be solved with exact algorithms, which are much faster in general; for example, the CPCS network can now be solved in less than 0.05 seconds with a personal computer (Díez and Galán, 2003). However, stochastic simulation can evaluate networks containing numeric variables, and for this reason it is still worth studying the basic algorithms.

Arc reversal is not the most efficient method for evaluating BNs—variable elimination is slightly faster and occupies the same amount of memory, and clustering is much faster for multiple queries at the cost of needing more memory. However, it is still one of the best algorithms for evaluating influence diagrams, mainly because it usually finds better elimination orderings than variable elimination and clustering (Luque and Díez, 2010). For this reason we teach our students this algorithm, first for BNs and then for influence diagrams.

With respect to learning BNs, OpenMarkov only implements the two basic algorithms and six metrics, but it has been carefully designed so that other researchers can add new methods and integrate them in the GUI for interactive learning.

Additionally, OpenMarkov offers the important advantage of being open-source, which means that the students with some knowledge of Java can inspect the implementation of the algorithms. For example, in the abstract class StochasticPropagation.java the students can find the data structures and methods common to the two algorithms discussed in this paper, while the classes that extend it, namely LogicSampling.java and LikelihoodWeighting.java, implement the aspects in which the algorithms differ.

Furthermore, advanced students can add new features to OpenMarkov—see for example (Li et al., 2018). In fact, a significant part of OpenMarkov's code has been written by our undergraduate, master, and PhD students.

5. Conclusion and future work

The facilities that OpenMarkov offers for teaching BNs are based on three features that, to our knowledge, are not available in any other tool: showing several probability bars simultaneously for different evidence cases, storing in a spreadsheet the samples generated by stochastic algorithms, and learning BNs interactively. With them we have been able to explain our students in an intuitive way some concepts related with conditional (in)dependence that we found difficult to explain when we did not have them. The possibility of learning BNs interactively using the two basic algorithms and several metrics for search-and-score has allowed our students to "play" with different databases and observe how the methods explained in the theory work in practice. The inversion of links at the GUI, which shows the new probability tables and the links added, may help understand the arc-reversal algorithm. Given that nowadays PGMs make part of the computer science curriculum in every university, we expect that many scholars around the world may consider OpenMarkov a useful tool for teaching them.

In the future it would be useful to implement in OpenMarkov new algorithms for inference and learning and new explanation facilities. We will extend this paper by describing some features that help us teach not only BNs but also influence diagrams using this tool.

Acknowledgments

This work has been supported by grant TIN2016-77206-R of the Spanish Government, co-financed by the European Regional Development Fund. I.P. received a grant from the Comunidad de Madrid, financed by the Youth Employment Initiative (YEI) of the European Social Fund. J.P. received a predoctoral grant from the Spanish Ministry of Education. We thank the reviewers of the PGM conference for many useful comments and corrections.

References

M. Arias, F. J. Díez, M. A. Palacios-Alonso, and I. Bermejo. ProbModelXML. A format for encoding probabilistic graphical models. In A. Cano, M. Gómez, and T. D. Nielsen, editors, *Proceedings of the Sixth European Workshop on Probabilistic Graphical Models (PGM'12)*, pages 11–18, Granada, Spain, 2012.

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

F. J. Díez and M. J. Druzdzel. Canonical probabilistic models for knowledge engineering. Technical Report CISIAD-06-01, UNED, Madrid, Spain, 2006.

F. J. Díez and S. F. Galán. Efficient computation for the noisy MAX. *International Journal of Intelligent Systems*, 18:165–177, 2003.

F. J. Díez, M. Yebra, I. Bermejo, M. A. Palacios-Alonso, M. Arias, M. Luque, and J. Pérez-Martín. Markov influence diagrams: A graphical tool for cost-effectiveness analysis. *Medical Decision Making*, 37:183–195, 2017.

F. J. Díez, M. Luque, and I. Bermejo. Decision analysis networks. *International Journal of Approximate Reasoning*, 96:1–17, 2018.

Elvira Consortium. Elvira: An environment for creating and using probabilistic graphical models. In J. A. Gámez and A. Salmerón, editors, *Proceedings of the First European Workshop on Probabilistic Graphical Models (PGM'02)*, pages 1–11, Cuenca, Spain, 2002.

R. Fung and K. C. Chang. Weighing and integrating evidence for stochastic simulation in Bayesian networks. In P. Bonissone, M. Henrion, L. N. Kanal, and J. F. Lemmer, editors, *Uncertainty in Artificial Intelligence 6 (UAI'90)*, pages 209–219, Amsterdam, The Netherlands, 1990. Elsevier Science Publishers.

M. Henrion. Propagation of uncertainty by logic sampling in Bayes' networks. In R. D. Shachter, T. Levitt, L. N. Kanal, and J. F. Lemmer, editors, *Uncertainty in Artificial Intelligence 4 (UAI'88)*, pages 149–164, Amsterdam, The Netherlands, 1988. Elsevier Science Publishers.

R. A. Howard and J. E. Matheson. Influence diagrams. In R. A. Howard and J. E. Matheson, editors, *Readings on the Principles and Applications of Decision Analysis*, pages 719–762. Strategic Decisions Group, Menlo Park, CA, 1984.

D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, Cambridge, MA, 2009.

L. Li, O. Ramadan, and P. Schmidt. Improving visual cues for the interactive learning of Bayesian networks, 2018. URL `http://vis.berkeley.edu/courses/cs294-10-fa14/wiki/images/0/0a/Li_Ramadan_Schmidt_Paper.pdf`. Downloaded: 31 May 2018.

M. Luque and F. J. Díez. Variable elimination for influence diagrams with super-value nodes. *International Journal of Approximate Reasoning*, 51:615–631, 2010.

R. E. Neapolitan. *Probabilistic Reasoning in Expert Systems: Theory and Algorithms*. Wiley-Interscience, New York, 1990.

F. A. Oliehoek, M. T. J. Spaan, B. Terwijn, P. Robbel, and J. V. Messias. The MADP Toolbox: An open source library for planning and learning in (multi-)agent systems. *Journal of Machine Learning Research*, 18(89):1–5, 2017.

S. M. Olmsted. *On Representing and Solving Decision Problems*. PhD thesis, Dept. Engineering-Economic Systems, Stanford University, CA, 1983.

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

J. Pearl. *Causality. Models, Reasoning, and Inference*. Cambridge University Press, Cambridge, UK, 2000.

R. D. Shachter. Evaluating influence diagrams. *Operations Research*, 34:871–882, 1986.

P. Spirtes and C. Glymour. An algorithm for fast recovery of sparse causal graphs. *Social Science Computer Review*, 9:62–72, 1991.

On the Properties of MVR Chain Graphs

Mohammad Ali Javidian JAVIDIAN@EMAIL.SC.EDU

Marco Valtorta MGV@CSE.SC.EDU

Department of Computer Science & Engineering, University of South Carolina, Columbia, SC, 29201, USA.

Abstract

Depending on the interpretation of the type of edges, a chain graph can represent different relations between variables and thereby independence models. Three interpretations, known by the acronyms LWF, MVR, and AMP, are prevalent. Multivariate regression chain graphs (MVR CGs) were introduced by Cox and Wermuth in 1993. We review Markov properties for MVR chain graphs and propose an alternative local Markov property for them. Except for pairwise Markov properties, we show that for MVR chain graphs all Markov properties in the literature are equivalent for semi-graphoids. We derive a new factorization formula for MVR chain graphs which is more explicit than and different from the proposed factorizations for MVR chain graphs in the literature. Finally, we provide a summary table comparing different features of LWF, AMP, and MVR chain graphs.

Keywords: multivariate regression chain graph; Markov property; graphical Markov models; factorization of probability distributions; conditional independence; marginalization of causal latent variable models; compositional graphoids.

1. Introduction

A probabilistic graphical model is a probabilistic model for which a graph represents the conditional dependence structure between random variables. There are several classes of graphical models; Bayesian networks (BN), Markov networks, chain graphs, and ancestral graphs are commonly used (Lauritzen, 1996; Richardson and Spirtes, 2002). Chain graphs, which admit both directed and undirected edges, are a type of graphs in which there are no partially directed cycles. Chain graphs were introduced by Lauritzen, Wermuth and Frydenberg (Frydenberg, 1990; Lauritzen and Wermuth, 1989) as a generalization of graphs based on undirected graphs and directed acyclic graphs (DAGs). Later on Andersson, Madigan and Perlman introduced an alternative Markov property for chain graphs (Andersson et al., 1996). In 1993 (Cox and Wermuth, 1993), Cox and Wermuth introduced *multivariate regression chain graphs (MVR CGs)*.

Acyclic directed mixed graphs (ADMGs), also known as semi-Markov(ian) (Pearl, 2009) models contain directed ($\rightarrow$) and bi-directed ($\leftrightarrow$) edges subject to the restriction that there are no directed cycles (Richardson, 2003; Evans and Richardson, 2014). An ADMG that has no partially directed cycle is called a *multivariate regression chain graph*. In this paper we focus on the class of multivariate regression chain graphs and we discuss their Markov properties. The discussion preceding Theorem 6 provides strong motivation for the importance of MVR CGs. In the first decade of the 21st century, several Markov property (global, pairwise, block recursive, and so on) were introduced by authors and researchers (Richardson and Spirtes, 2002; Wermuth and Cox, 2004; Marchetti and Lupparelli, 2008, 2011; Drton, 2009). Lauritzen, Wermuth, and Sadeghi (Sadeghi and Lauritzen, 2014; Sadeghi and Wermuth, 2016) proved that the global and (four) pairwise Markov properties of

a MVR chain graph are equivalent for any independence model that is a compositional graphoid. The major contributions of this paper may be summarized as follows:
• Proposed an alternative local Markov property for MVR chain graphs, which is equivalent with other Markov properties in the literature for compositional semi-graphoids.
• Compared different proposed Markov properties for MVR chain graphs in the literature and considered conditions under which they are equivalent.
• Derived an alternative explicit factorization criterion for MVR chain graphs based on the proposed factorization criterion for acyclic directed mixed graphs in (Evans and Richardson, 2014).

2. Definitions and Concepts

Definition 1 *A vertex α is said to be an* ancestor *of a vertex β if either there is a directed path $\alpha \to \ldots \to \beta$ from α to β, or $\alpha = \beta$. A vertex α is said to be* anterior *to a vertex β if there is a path μ from α to β on which every edge is either of the form $\gamma - \delta$, or $\gamma \to \delta$ with δ between γ and β, or $\alpha = \beta$; that is, there are no edges $\gamma \leftrightarrow \delta$ and there are no edges $\gamma \leftarrow \delta$ pointing toward α. Such a path is said to be an anterior path from α to β. We apply these definitions disjunctively to sets: $an(X) = \{\alpha | \alpha$ is an ancestor of β for some $\beta \in X\}$, and $ant(X) = \{\alpha | \alpha$ is an anterior of β for some $\beta \in X\}$. If necessary we specify the graph by a subscript, as in $ant_G(X)$. The usage of the terms "ancestor" and "anterior" differs from Lauritzen (Lauritzen, 1996), but follows Frydenberg (Frydenberg, 1990).*

Definition 2 *A mixed graph is a graph containing three types of edges, undirected ($-$), directed ($\to$) and bidirected ($\leftrightarrow$). An ancestral graph G is a mixed graph in which the following conditions hold for all vertices α in G:*
(i) if α and β are joined by an edge with an arrowhead at α, then α is not anterior to β.
(ii) there are no arrowheads present at a vertex which is an endpoint of an undirected edge.

Definition 3 *A nonendpoint vertex ζ on a path is a* collider *on the path if the edges preceding and succeeding ζ on the path have an arrowhead at ζ, that is, $\to \zeta \leftarrow$, or $\leftrightarrow \zeta \leftrightarrow$, or $\leftrightarrow \zeta \leftarrow$, or $\to \zeta \leftrightarrow$. A nonendpoint vertex ζ on a path which is not a collider is a noncollider on the path. A path between vertices α and β in an ancestral graph G is said to be m-connecting given a set Z (possibly empty), with $\alpha, \beta \notin Z$, if:*
(i) every noncollider on the path is not in Z, and
(ii) every collider on the path is in $ant_G(Z)$.

If there is no path m-connecting α and β given Z, then α and β are said to be m-separated *given Z. Sets X and Y are m-separated given Z, if for every pair α, β, with $\alpha \in X$ and $\beta \in Y$, α and β are m-separated given Z (X, Y, and Z are disjoint sets; X, Y are nonempty). This criterion is referred to as a* global Markov property. *We denote the independence model resulting from applying the m-separation criterion to G, by $\Im_m$(G). This is an extension of Pearl's d-separation criterion to mixed graphs in that in a DAG D, a path is d-connecting if and only if it is m-connecting.*

Definition 4 *Let G_A denote the induced subgraph of G on the vertex set A, formed by removing from G all vertices that are not in A, and all edges that do not have both endpoints in A. Two vertices x and y in a MVR chain graph G are said to be collider connected if there is a path from x to y in G on which every non-endpoint vertex is a collider; such a path is called a collider path. (Note that a single edge trivially forms a collider path, so if x and y are adjacent in a MVR chain graph then*

they are collider connected.) The augmented graph *derived from G, denoted* $(G)^a$*, is an undirected graph with the same vertex set as G such that* $c - d$ *in* $(G)^a \Leftrightarrow$ *c and d are collider connected in* G.

Definition 5 *Disjoint sets* $X, Y \neq \emptyset$*, and Z (Z may be empty) are said to be* m^*-separated *if X and Y are separated by Z in* $(G_{ant(X \cup Y \cup Z)})^a$*. Otherwise X and Y are said to be* m^**-connected given Z. The resulting independence model is denoted by* $\Im_{m^*}(G)$.

Richardson and Spirtes in (Richardson and Spirtes, 2002, Theorem 3.18.) show that for an ancestral graph G, $\Im_m(G) = \Im_{m^*}(G)$. Note that in the case of ADMGs and MVR CGs, anterior sets in definitions 3, 5 can be replaced by ancestor sets, because in both cases anterior sets and ancestor sets are the same.

The absence of partially directed cycles in MVR CGs implies that the vertex set of a chain graph can be partitioned into so-called chain components such that edges within a chain component are bidirected whereas the edges between two chain components are directed and point in the same direction. So, any chain graph yields a directed acyclic graph D of its chain components having $\mathcal{T}$ as a node set and an edge $T_1 \rightarrow T_2$ whenever there exists in the chain graph G at least one edge $u \rightarrow v$ connecting a node *u* in T_1 with a node *v* in T_2. In this directed graph, we may define for each T the set $pa_D(T)$ as the union of all the chain components that are parents of T in the directed graph D. This concept is distinct from the usual notion of the parents $pa_G(A)$ of a set of nodes A in the chain graph, that is, the set of all the nodes w outside A such that $w \rightarrow v$ with $v \in A$ (Marchetti and Lupparelli, 2011). Given a chain graph G with chain components $(T|T \in \mathcal{T})$, we can always define a strict total order $\prec$ of the chain components that is consistent with the partial order induced by the chain graph, such that if $T \prec T'$ then $T \notin pa_D(T')$ (we draw T' to the right of T as in the example of Figure 1). For each T, the set of all components preceding T is known and we may define the cumulative set $pre(T) = \cup_{T \prec T'} T'$ of nodes contained in the predecessors of component T, which we sometimes call the past of T. The set $pre(T)$ captures the notion of all the potential explanatory variables of the response variables within T (Marchetti and Lupparelli, 2011).

3. Markov Properties for MVR Chain Graphs

In this section, first, we show, formally, that MVR chain graphs are a subclass of the maximal ancestral graphs of Richardson and Spirtes (Richardson and Spirtes, 2002) that include only observed and latent variables. Latent variables cause several complications (Colombo et al., 2012). First, causal inference based on structural learning algorithms such as the PC algorithm (Spirtes et al., 2000) may be incorrect. Second, if a distribution is faithful to a DAG, then the distribution obtained by marginalizing out on some of the variables may not be faithful to any DAG on the observed variables i.e., the space of DAGs is not closed under marginalization. These problems can be solved by exploiting MVR chain graphs. This motivates the development of studies on MVR CGs.

Theorem 6 *If G is a MVR chain graph, then G is an ancestral graph.*

Proof Obviously, every MVR chain graph is a mixed graph without undirected edges. So, it is enough to show that condition (i) in Definition 2 is satisfied. For this purpose, consider that α and β are joined by an edge with an arrowhead at α in MVR chain graph G. Two cases are possible. First, if $\alpha \leftrightarrow \beta$ is an edge in G, by definition of a MVR chain graph, both of them belong to the same chain component. Since all edges on a path between two nodes of a chain component are bidirected,

then by definition α cannot be an anterior of β. Second, if $\alpha \leftarrow \beta$ is an edge in G, by definition of a MVR chain graph, α and β belong to two different components (β is in a chain component that is to the right side of the chain component that contains α). We know that all directed edges in a MVR chain graph are arrows pointing from right to left, so there is no path from α to β in G i.e. α cannot be an anterior of β in this case. We have shown that α cannot be an anterior of β in both cases, and therefore condition (i) in Definition 2 is satisfied. In other words, every MVR chain graph is an ancestral graph. ■

The following result is often mentioned in the literature (Wermuth and Sadeghi, 2012; Peña, 2015; Sadeghi and Lauritzen, 2014; Sonntag, 2014), but we know of no published proof.

Corollary 7 *Every MVR chain graph has the same independence model as a DAG under marginalization.*

Proof From Theorem 6, we know that every MVR chain graph is an ancestral graph. The result follows directly from (Richardson and Spirtes, 2002, Theorem 6.3). ■

3.1 Global and Pairwise Markov Properties

The following properties have been defined for conditional independences of probability distributions. Let A, B, C and D be disjoint subsets of V_G, where C may be the empty set.
1. Symmetry: $A \perp\!\!\!\perp B \Rightarrow B \perp\!\!\!\perp A$;
2. Decomposition: $A \perp\!\!\!\perp BD|C \Rightarrow (A \perp\!\!\!\perp B|C$ and $A \perp\!\!\!\perp D|C)$;
3. Weak union: $A \perp\!\!\!\perp BD|C \Rightarrow (A \perp\!\!\!\perp B|DC$ and $A \perp\!\!\!\perp D|BC)$;
4. Contraction: $(A \perp\!\!\!\perp B|DC$ and $A \perp\!\!\!\perp D|C) \Leftrightarrow A \perp\!\!\!\perp BD|C$;
5. Intersection: $(A \perp\!\!\!\perp B|DC$ and $A \perp\!\!\!\perp D|BC) \Rightarrow A \perp\!\!\!\perp BD|C$;
6. Composition: $(A \perp\!\!\!\perp B|C$ and $A \perp\!\!\!\perp D|C) \Rightarrow A \perp\!\!\!\perp BD|C$.

An independence model is a *semi-graphoid* if it satisfies the first four independence properties listed above. Note that every probability distribution p satisfies the *semi-graphoid* properties (Studený, 1989). If a semi-graphoid further satisfies the intersection property, we say it is a *graphoid* (Pearl and Paz, 1987; Studený, 2005, 1989). A *compositional graphoid* further satisfies the composition property (Sadeghi and Wermuth, 2016). If a semi-graphoid further satisfies the composition property, we say it is a *compositional semi-graphoid*.

For a node i in the connected component T, its *past*, denoted by $pst(i)$, consists of all nodes in components having a higher order than T. To define pairwise Markov properties for MVR CGs, we use the following notation for parents, anteriors and the past of node pair i, j: $pa_G(i,j) = pa_G(i) \cup pa_G(j) \setminus \{i,j\}$, $ant(i,j) = ant(i) \cup ant(j) \setminus \{i,j\}$, and $pst(i,j) = pst(i) \cup pst(j) \setminus \{i,j\}$. The distribution $\mathcal{P}$ of $(X_n)_{n \in V}$ satisfies a pairwise Markov property (Pm), for $m = 1, 2, 3, 4$, with respect to MVR CG(G) if for every uncoupled pair of nodes i and j (i.e., there is no directed or bidirected edge between i and j):
(P1): $i \perp\!\!\!\perp j|pst(i,j)$, (P2): $i \perp\!\!\!\perp j|ant(i,j)$, (P3): $i \perp\!\!\!\perp j|pa_G(i,j)$, and (P4): $i \perp\!\!\!\perp j|pa_G(i)$ if $i \prec j$.

Notice that in (P4), $pa_G(i)$ may be replaced by $pa_G(j)$ whenever the two nodes are in the same connected component. Sadeghi and Wermuth in (Sadeghi and Wermuth, 2016) proved that all of above mentioned pairwise Markov properties are equivalent for compositional graphoids. Also,

they show that each one of the above listed pairwise Markov properties is equivalent to the global Markov properties in Definitions 3, 5 (Sadeghi and Wermuth, 2016, Corollary 1). The necessity of intersection and composition properties follows from (Sadeghi and Lauritzen, 2014, Section 6.3).

3.2 Block-recursive, Multivariate Regression (MR), and Ordered Local Markov Properties

Definition 8 *Given a chain graph G, the set $Nb_G(A)$ is the union of A itself and the set of nodes w that are neighbors of A, that is, coupled by a bi-directed edge to some node v in A. Moreover, the set of non-descendants $nd_D(T)$ of a chain component T, is the union of all components T' such that there is no directed path from T to T' in the directed graph of chain components D.*

Definition 9 (multivariate regression (MR) Markov property for MVR CGs (Marchetti and Lupparelli, 2011)) *Let G be a chain graph with chain components $(T|T \in \mathcal{T})$. A joint distribution P of the random vector X obeys* multivariate regression (MR) Markov property *with respect to G if it satisfies the following independences. For all $T \in \mathcal{T}$ and for all $A \subseteq T$:*
(MR1) if A is connected: $A \perp\!\!\!\perp [pre(T) \setminus pa_G(A)]|pa_G(A)$.
(MR2) if A is disconnected with connected components $A_1, \ldots, A_r$: $A_1 \perp\!\!\!\perp \ldots \perp\!\!\!\perp A_r|pre(T)$.

Remark 10 *(Marchetti and Lupparelli, 2011, Remark 2) One immediate consequence of Definition 9 is that if the probability density p(x) is strictly positive, then it factorizes according to the directed acyclic graph of the chain components: $p(x) = \prod_{T\in\mathcal{T}} p(x_T|x_{pa_D(T)})$.*

Definition 11 (Chain graph Markov property of type IV (Drton, 2009)) *Let G be a chain graph with chain components $(T|T \in \mathcal{T})$ and directed acyclic graph D of components. The joint probability distribution of X obeys* the block-recursive Markov property of type IV *if it satisfies the following independencies:*
(IV0): $T \perp\!\!\!\perp [nd_D(T) \setminus pa_D(T)]|pa_D(T)$, for all $T \in \mathcal{T}$;
(IV1): $A \perp\!\!\!\perp [pa_D(T) \setminus pa_G(A)]|pa_G(A)$, for all $T \in \mathcal{T}$, and for all $A \subseteq T$;
(IV2): $A \perp\!\!\!\perp [T \setminus Nb_G(A)]|pa_D(T)$, for all $T \in \mathcal{T}$, and for all connected subsets $A \subseteq T$.

The following example shows that independence models, in general, resulting from Definitions 9, 11 are different.

Example 1 *Consider the MVR chain graph G in Figure 1. For the connected set $A = \{1, 2\}$ the*

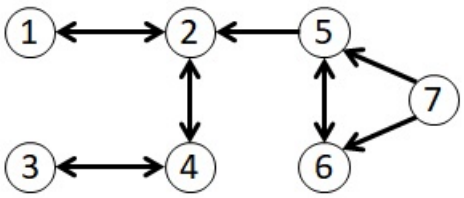


Figure 1: A MVR CG with chain components: $\mathcal{T} = \{T_1 = \{1, 2, 3, 4\}, T_2 = \{5, 6\}, T_3 = \{7\}\}$.

condition (MR1) implies that $1, 2 \perp\!\!\!\perp 6, 7|5$ while the condition (IV2) implies that $1, 2 \perp\!\!\!\perp 6|5$, which is not implied directly by (MR1) and (MR2). Also, the condition (MR2) implies that $1 \perp\!\!\!\perp 3, 4|5, 6, 7$ while the condition (IV2) implies that $1 \perp\!\!\!\perp 3, 4|5, 6$, which is not implied directly by (MR1) and (MR2).

Theorem 1 in (Marchetti and Lupparelli, 2011) states that for a given chain graph G, the multivariate regression Markov property is equivalent to the block-recursive Markov property of type IV. Also, Drton in (Drton, 2009, Section 7 Discussion) claims that (without proof) the block-recursive Markov property of type IV can be shown to be equivalent to the global Markov property proposed in (Richardson and Spirtes, 2002; Richardson, 2003).

Now, we introduce a(n ordered) local Markov property for ADMGs proposed by Richardson in (Richardson, 2003), which is an extension of the local well-numbering Markov property for DAGs introduced in (Lauritzen et al., 1990). For this purpose, we need to consider the following definitions and notations:

Definition 12 *For a given acyclic directed mixed graph (ADMG) G, the induced bi-directed graph $(G)_{\leftrightarrow}$ is the graph formed by removing all directed edges from G. The district (aka c-component) for a vertex x in G is the connected component of x in $(G)_{\leftrightarrow}$, or equivalently*

$$dis_G(x) = \{y | y \leftrightarrow \ldots \leftrightarrow x \text{ in } G, \text{ or } x = y\}.$$

As usual we apply the definition disjunctively to sets: $dis_A(B) = \cup_{x \in B} dis_A(x)$. A set C is path-connected in $(G)_{\leftrightarrow}$ if every pair of vertices in C are connected via a path in $(G)_{\leftrightarrow}$; equivalently, every vertex in C has the same district in G.

Definition 13 *In an ADMG, a set A is said to be ancestrally closed if $x \rightarrow \ldots \rightarrow a$ in G with $a \in A$ implies that $x \in A$. The set of ancestrally closed sets is defined as follows:*

$$\mathcal{A}(G) = \{A | an_G(A) = A\}.$$

If A is an ancestrally closed set in an ADMG (G), and x is a vertex in A that has no children in A then we define the Markov blanket *of a vertex x with respect to the induced subgraph on A as*

$$mb(x, A) = pa_G(dis_{G_A}(x)) \cup (dis_{G_A}(x) \setminus \{x\}),$$

where dis_{G_A} is the district of x in the induced subgraph G_A.

Definition 14 *Let G be an acyclic directed mixed graph. Specify a total ordering ($\prec$) on the vertices of G, such that $x \prec y \Rightarrow y \notin an(x)$; such an ordering is said to be consistent with G. Define $pre_{G,\prec}(x) = \{v | v \prec x \text{ or } v = x\}$.*

Definition 15 (Ordered local Markov property) *Let G be an acyclic directed mixed graph. An independence model $\Im$ over the node set of G satisfies the ordered local Markov property for G, with respect to the ordering $\prec$, if for any x, and ancestrally closed set A such that $x \in A \subseteq pre_{G,\prec}(x)$,*

$$\{x\} \perp\!\!\!\perp [A \setminus (mb(x, A) \cup \{x\})] | mb(x, A).$$

Since MVR chain graphs are a subclass of ADMGs, the ordered local Markov property in Definition 15 can be used as a local Markov property for MVR chain graphs.

Theorem 16 *Let G be a MVR chain graph. For an independence model $\Im$ over the node set of G, the following conditions are equivalent:*
(i) $\Im$ satisfies the global Markov property w.r.t. G in Definition 3;
(ii) $\Im$ satisfies the global Markov property w.r.t. G in Definition 5;
(iii) $\Im$ satisfies the block recursive Markov property w.r.t. G in Definition 11;
(iv) $\Im$ satisfies the MR Markov property w.r.t. G in Definition 9.
(v) $\Im$ satisfies the ordered local Markov property w.r.t. G in Definition 15.

Proof The proof of this theorem is omitted to save space; it is contained in the supplementary material (Javidian and Valtorta, 2018a). ■

3.3 An Alternative Local Markov Property for MVR Chain Graphs

In this subsection we formulate an alternative local Markov property for MVR chain graphs. This property is different from and much more concise than the ordered Markov property proposed in (Richardson, 2003). The new local Markov property can be used to parameterize distributions efficiently when MVR chain graphs are learned from data, as done, for example, in (Javidian and Valtorta, 2018b, Lemma 9). We show that this local Markov property is equivalent to the global and ordered local Markov property for MVR chain graphs (for compositional graphoids).

Definition 17 *If there is a bidirected edge between vertices u and v, u and v are said to be neighbors. The boundary $bd(u)$ of a vertex u is the set of vertices in $V \setminus \{u\}$ that are parents or neighbors of vertex u. The descendants of vertex u are $de(u) = \{v | u \text{ is an ancestor of } v\}$. The non-descendants of vertex u are $nd(u) = V \setminus (de(u) \cup \{u\})$.*

Definition 18 *The local Markov property for a MVR chain graph G with vertex set V holds if, for every $v \in V$: $v \perp\!\!\!\perp [nd(v) \setminus bd(v)] | pa_G(v)$.*

Remark 19 *In DAGs, $bd(v) = pa_G(v)$, and the local Markov property given above reduces to the directed local Markov property introduced by Lauritzen et al. in (Lauritzen et al., 1990). Also, in covariance graphs the local Markov property given above reduces to the dual local Markov property introduced by Kauermann in (Kauermann, 1996, Definition 2.1).*

Theorem 20 *Let G be a MVR chain graph. If an independence model $\Im$ over the node set of G is a compositional semi-graphoid, then $\Im$ satisfies the alternative local Markov property w.r.t. G in Definition 18 if and only if it satisfies the global Markov property w.r.t. G in Definition 5.*

Proof $(Global \Rightarrow Local)$: Let $X = \{v\}, Y = nd(v) \setminus bd(v),$ and $Z = pa_G(v)$. So, $an(X \cup Y \cup S) = v \cup (nd(v) \setminus bd(v)) \cup pa_G(v)$ is an ancestor set, and $pa_G(v)$ separates v from $nd(v) \setminus bd(v)$ in $(G_{v \cup (nd(v) \setminus bd(v)) \cup pa_G(v)})^a$; this shows that the global Markov property in Definition 5 implies the local Markov property in Definition 18.
$(Local \Rightarrow MR)$: We prove this by considering the following two cases:
Case 1): Let $A \subseteq T$ is connected. Using the alternative local Markov property for each $x \in A$ implies that: $\{x\} \perp\!\!\!\perp [nd(x) \setminus bd(x)] | pa_G(x)$. Since $(pre(T) \setminus pa_G(A)) \subseteq (nd(x) \setminus bd(x))$, using the decomposition and weak union property give: $\{x\} \perp\!\!\!\perp (pre(T) \setminus pa_G(A)) | pa_G(A)$, for all $x \in A$. Using the composition property leads to (MR1): $A \perp\!\!\!\perp (pre(T) \setminus pa_G(A)) | pa_G(A)$.
Case 2): Let $A \subseteq T$ is disconnected with connected components $A_1, \ldots, A_r$. For $1 \leq i \neq j \leq r$ we have: $\{x\} \perp\!\!\!\perp [nd(x) \setminus bd(x)] | pa_G(x)$, for all $x \in A_i$. Since $[(pre(T) \setminus pa_G(A)) \cup A_j] \subseteq (nd(x) \setminus bd(x))$, using the decomposition and weak union property give: $\{x\} \perp\!\!\!\perp A_j | pre(T)$, for all $x \in A_i$. Using the composition property leads to (MR2): $A_i \perp\!\!\!\perp A_j | pre(T),$ for all $1 \leq i \neq j \leq r$.
$(MR \Rightarrow Global)$: The result follows from Theorem 16. ■

The necessity of composition property in Theorem 20 follows from the fact that local and global Markov properties for bi-directed graphs, which are a subclass of MVR CGs, are equivalent only for compositional semi-graphoids (Kauermann, 1996, Proposiotion 2.2).

4. An Alternative Factorization for MVR Chain Graphs

According to the definition of MVR chain graphs, it is obvious that they are a subclass of acyclic directed mixed graphs (ADMGs). In this section, we derive an explicit factorization criterion for MVR chain graphs based on the proposed factorization criterion for acyclic directed mixed graphs in (Evans and Richardson, 2014). For this purpose, we need to consider the following definition and notations:

Definition 21 *An ordered pair of sets (H, T) form the head and tail of a term associated with an ADMG G if and only if all of the following hold:*
1. $H = barren(H)$, where $barren(H) = \{v \in H | de(v) \cap H = \{v\}\}$.
2. H contained within a single district of $G_{an(H)}$.
3. $T = tail(H) = (dis_{an(H)}(H) \setminus H) \cup pa(dis_{an(H)}(H))$.

Evans and Richardson in (Evans and Richardson, 2014, Theorem 4.12) prove that a probability distribution P obeys the global Markov property for an ADMG(G) if and only if for every $A \in \mathcal{A}(G)$,

$$p(X_A) = \prod_{H \in [A]_G} p(X_H | tail(H)), \tag{1}$$

where $[A]_G$ denotes a partition of A into sets $\{H_1, \ldots, H_k\} \subseteq \mathcal{H}(G)$ (for a graph G, the set of heads is denoted by $\mathcal{H}(G)$), defined with $tail(H)$, as above. The following theorem provides an alternative factorization criterion for MVR chain graphs based on the proposed factorization criterion for acyclic directed mixed graphs in (Evans and Richardson, 2014).

Theorem 22 *Let G be a MVR chain graph with chain components $(T | T \in \mathcal{T})$. If a probability distribution P obeys the global Markov property for G then $p(x) = \prod_{T \in \mathcal{T}} p(x_T | x_{pa_G(T)})$.*

Proof According to Theorem 4.12 in (Evans and Richardson, 2014), since $G \in \mathcal{A}(G)$, it is enough to show that $\mathcal{H}(G) = \{T | T \in \mathcal{T}\}$ and $tail(T) = pa_G(T)$, where $T \in \mathcal{T}$. In other words, it is enough to show that for every T in $\mathcal{T}$, $(T, pa_G(T))$ satisfies the three conditions in Definition 21.
1. Let $x, y \in T$ and $T \in \mathcal{T}$. Then y is not a descendant of x. Also, we know that $x \in de(x)$, by definition. Therefore, $T = barren(T)$.
2. Let $T \in \mathcal{T}$, then from the definitions of a MVR chain graph and induced bi-directed graph, it is obvious that T is a single connected component of the forest $(G_{an(T)})_{\leftrightarrow}$. So, T contained within a single district of $(G_{an(T)})_{\leftrightarrow}$.
3. $T \subseteq an(T)$ by definition. So, $\forall x \in T : dis_{an(T)}(x) = \{y | y \leftrightarrow \ldots \leftrightarrow x \text{ in } an(T), \text{or } x = y\} = T$. Therefore, $dis_{an(T)}(T) = T$ and $dis_{an(T)}(T) \setminus T = \emptyset$. In other words, $tail(T) = pa_G(T)$. ■

Example 2 *Consider the MVR chain graph G in Example 1. Since $[G]_G = \{\{1, 2, 3, 4\}\{5, 6\}\{7\}\}$ so, $tail(\{1, 2, 3, 4\}) = \{5\}, tail(\{5, 6\}) = \{7\}$, and $tail(\{7\}) = \emptyset$. Therefore, based on Theorem 22 we have: $p = p_{1234|5} p_{56|7} p_7$. However, the corresponding factorization of G based on the formula in (Drton, 2009; Marchetti and Lupparelli, 2011) is: $p = p_{1234|56} p_{56|7} p_7$.*

The advantage of the new factorization is that it requires only graphical parents, rather than parent components in each factor, resulting in smaller variable sets for each factor, and therefore speeding up belief propagation.

Type of chain graph	Does it represent independence model of DAGs under marginalization?	Global Markov property	Factorization of $p(x)$	Model selection (structural learning) algorithm(s) [constraint based method]
MVR CGs: Cox & Wermuth (Cox and Wermuth, 1993, 1996; Wermuth and Cox, 2004), Peña & Sonntag (Peña, 2015; Sonntag, 2014), Sadeghi & Lauritzen (Sadeghi and Lauritzen, 2014), Drton (type IV) (Drton, 2009), Marchetti & Lupparelli (Marchetti and Lupparelli, 2008, 2011)	Yes (claimed in (Cox and Wermuth, 1996; Wermuth and Sadeghi, 2012; Sadeghi and Lauritzen, 2014; Sonntag, 2014), proved in Corollary 7)	(1) $X \perp\!\!\!\perp Y \mid Z$ if X is separated from Y by Z in $(G_{ant(X\cup Y\cup Z)})^a$ or $(G_{an(X\cup Y\cup Z)})^a$ (Richardson, 2003; Richardson and Spirtes, 2002). (2) $X \perp\!\!\!\perp Y \mid Z$ if X is separated from Y by Z in $(G_{Antec(X\cup Y\cup Z)})^a$. (1) and (2) are equivalent for compositional graphoids (see supplementary material).	(1) Theorem 22, $\prod_{T\in\mathcal{T}} p(x_T \mid x_{pa(T)})$ (2) $\prod_{T\in\mathcal{T}} p(x_T \mid x_{pa_D(T)})$ where $pa_D(T)$ is the union of all the chain components that are parents of T in the directed graph D (Drton, 2009; Marchetti and Lupparelli, 2011).	PC like algorithm for MVR CGs in (Sonntag, 2014; Sonntag and Peña, 2012), Decomposition-based algorithm for MVR CGs in (Javidian and Valtorta, 2018b).
LWF CGs (Frydenberg, 1990; Lauritzen and Wermuth, 1989), Drton (type I) (Drton, 2009)	No	$X \perp\!\!\!\perp Y \mid Z$ if X is separated from Y by Z in $(G_{An(X\cup Y\cup Z)})^m$ (Lauritzen, 1996).	(Cowell et al., 1999; Lauritzen and Richardson, 2002) $\prod_{\tau\in\mathcal{T}} p(x_\tau \mid x_{pa(\tau)})$, where $p(x_\tau \mid x_{pa(\tau)}) = Z^{-1}(x_{pa(\tau)}) \prod_{c\in C} \phi_c(x_c)$, where C are the complete sets in the moral graph $(\tau \cup pa(\tau))^m$.	PC like algorithm in (Studený, 1997), LCD algorithm in (Ma et al., 2008), CKES algorithm in (Peña et al., 2014; Sonntag, 2014)
AMP CGs (Andersson et al., 1996), Drton (type II) (Drton, 2009)	No	$X \perp\!\!\!\perp Y \mid Z$ if X is separated from Y by Z in the undirected graph $Aug[CG; X, Y, Z]$ (Richardson, 1998).	$\prod_{\tau\in\mathcal{T}} p(x_\tau \mid x_{pa(\tau)})$, where no further factorization similar to LWF model appears to hold in general (Andersson et al., 1996). For the positive distribution p see (Peña, 2018).	PC like algorithm in (Peña, 2014)

Table 1: Properties of chain graphs under different interpretations

Conclusion and Summary

Based on the interpretation of the type of edges in a chain graph, there are different conditional independence structures among random variables in the corresponding probabilistic model. Other than pairwise Markov properties, we showed that for MVR chain graphs all Markov properties in the literature are equivalent for semi-graphoids. We proposed an alternative local Markov property for MVR chain graphs, and we proved that it is equivalent with other Markov properties for compositional semi-graphoids. Also, we obtained an alternative formula for factorization of a MVR chain graph. Table 1 summarizes some of the most important attributes of different types of common interpretations of chain graphs.

Acknowledgements

This work has been partially supported by Office of Naval Research grant ONR N00014-17-1-2842. This research is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), award/contract number 2017-16112300009. The views and conclusions contained therein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes, notwithstanding annotation therein. Heartfelt thanks to the anonymous reviewers for excellent suggestions.

References

S. A. Andersson, D. Madigan, and M. D. Perlman. Alternative markov properties for chain graphs. *Uncertainty in artificial intelligence*, pages 40–48, 1996.

D. Colombo, M. H. Maathuis, M. Kalisch, and T. S. Richardson. Learning high-dimensional directed acyclic graphs with latent and selection variables. *The Annals of Statistics*, 40(1):294–321, 2012.

R. Cowell, A. P. Dawid, S. Lauritzen, and D. J. Spiegelhalter. *Probabilistic networks and expert systems. Statistics for Engineering and Information Science*. Springer-Verlag, 1999.

D. R. Cox and N. Wermuth. Linear dependencies represented by chain graphs. *Statistical Science*, 8(3):204–218, 1993.

D. R. Cox and N. Wermuth. *Multivariate Dependencies-Models, Analysis and Interpretation*. Chapman and Hall, 1996.

M. Drton. Discrete chain graph models. *Bernoulli*, 15(3):736–753, 2009.

R. Evans and T. S. Richardson. Markovian acyclic directed mixed graphs for discrete data. *The Annals of Statistics*, 42(4):1452–1482, 2014.

M. Frydenberg. The chain graph markov property. *Scandinavian Journal of Statistics*, 17(4):333–353, 1990.

M. A. Javidian and M. Valtorta. On the properties of MVR chain graphs. *https://arxiv.org/abs/1803.04262*, 2018a.

M. A. Javidian and M. Valtorta. Structural learning of multivariate regression chain graphs via decomposition. *https://arxiv.org/abs/1806.00882*, 2018b.

G. Kauermann. On a dualization of graphical gaussian models. *Scandinavian Journal of Statistics*, 23(1):105–116, 1996.

S. Lauritzen. *Graphical Models*. Oxford Science Publications, 1996.

S. Lauritzen and T. Richardson. Chain graph models and their causal interpretations. *Journal of the Royal Statistical Society. Series B, Statistical Methodology*, 64(3):321–348, 2002.

S. Lauritzen and N. Wermuth. Graphical models for associations between variables, some of which are qualitative and some quantitative. *The Annals of Statistics*, 17(1):31–57, 1989.

S. Lauritzen, A. P. Dawid, B. N. Larsen, and H.-G. Leimer. Independence properties of directed markov fields. *Networks*, 20(5):491–505, 1990.

Z. Ma, X. Xie, and Z. Geng. Structural learning of chain graphs via decomposition. *Journal of Machine Learning Research*, 9:2847–2880, 2008.

G. Marchetti and M. Lupparelli. Parameterization and fitting of a class of discrete graphical models. *COMPSTAT: Proceedings in Computational Statistics. P. Brito. Heidelberg, Physica-Verlag HD*, pages 117–128, 2008.

G. Marchetti and M. Lupparelli. Chain graph models of multivariate regression type for categorical data. *Bernoulli*, 17(3):827–844, 2011.

J. Pearl. *Causality. Models, reasoning, and inference*. Cambridge University Press, 2009.

J. Pearl and A. Paz. Graphoids: a graph based logic for reasoning about relevancy relations. *Advances in Artificial Intelligence II Boulay, BD, Hogg, D & Steel, L (eds), North Holland, Amsterdam*, pages 357–363, 1987.

J. M. Peña. Learning marginal AMP chain graphs under faithfulness. *European Workshop on Probabilistic Graphical Models PGM: Probabilistic Graphical Models*, pages 382–395, 2014.

J. M. Peña. Every LWF and AMP chain graph originates from a set of causal models. *Symbolic and quantitative approaches to reasoning with uncertainty, Lecture Notes in Comput. Sci., 9161, Lecture Notes in Artificial Intelligence, Springer, Cham*, pages 325–334, 2015.

J. M. Peña. Reasoning with alternative acyclic directed mixed graphs. *Behaviormetrika*, pages 1–34, 2018.

J. M. Peña, D. Sonntag, and J. Nielsen. An inclusion optimal algorithm for chain graph structure learning. *In Proceedings of the 17th International Conference on Artificial Intelligence and Statistics*, pages 778–786, 2014.

T. S. Richardson. Chain graphs and symmetric associations. *In: Jordan M.I. (eds) Learning in Graphical Models. NATO ASI Series (Series D: Behavioural and Social Sciences), vol 89*, pages 229–259, 1998.

T. S. Richardson. Markov properties for acyclic directed mixed graphs. *Scandinavian Journal of Statistics*, 30(1):145–157, 2003.

T. S. Richardson and P. Spirtes. Ancestral graph markov models. *The Annals of Statistics*, 30(4): 962–1030, 2002.

K. Sadeghi and S. Lauritzen. Markov properties for mixed graphs. *Bernoulli*, 20(2):676–696, 2014.

K. Sadeghi and N. Wermuth. Pairwise markov properties for regression graphs. *Stat*, 5:286–294, 2016.

D. Sonntag. *A Study of Chain Graph Interpretations (Licentiate dissertation) [https://doi.org/10.3384/lic.diva-105024]*. Linköping University, 2014.

D. Sonntag and J. M. Peña. Learning multivariate regression chain graphs under faithfulness. *In: Proceedings of the 6th European Workshop on Probabilistic Graphical Models*, pages 299–306, 2012.

P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction and Search, second ed.* MIT Press, Cambridge, MA., 2000.

M. Studený. Multiinformation and the problem of characterization of conditional independence relations. *Problems of Control and Information Theory*, 18:3–16, 1989.

M. Studený. A recovery algorithm for chain graphs. *International Journal of Approximate Reasoning*, 17:265–293, 1997.

M. Studený. *Probabilistic Conditional Independence Structures*. Springer-Verlag London, 2005.

N. Wermuth and D. R. Cox. Joint response graphs and separation induced by triangular systems. *Journal of the Royal Statistical Society. Series B, Statistical Methodology*, 66(3):687–717, 2004.

N. Wermuth and K. Sadeghi. Sequences of regressions and their independences. *Test*, 21:215–252, 2012.

Variation Intervals for Posterior Probabilities in Bayesian Networks in Anticipation of Future Observations

Marcin Kozniewski MAK295@PITT.EDU

Marek J. Druzdzel DRUZDZEL@PITT.EDU

Decision Systems Laboratory, School of Computing and Information, University of Pittsburgh, 135 N Bellefield, PA 15260, USA

Faculty of Computer Science, Białystok University of Technology, Wiejska 45A, 15-351 Białystok, Poland

Abstract

Most applications of Bayesian networks focus on calculating posterior probabilities over variables of interest given observations of other variables. Because not all observations are available at the outset, one would like to know how future observations may lead to changes of these posterior probabilities. For example, a probability of a disease in a patient with little or no symptoms or test results is close to disease prevalence in general population. This probability can go up or down, depending on the patient's specifics. A user of a probabilistic decision support system might want to know where this probability can go as more information becomes available. We propose to address this problem by deriving variation intervals over posterior probabilities. Our method involves simulation of future observations to calculate possible values of posterior probabilities.

Keywords: Bayesian networks, uncertainty, posterior probability, variation intervals, confidence intervals, information gathering, simulation

Bayesian network (BN) Pearl (1988) is a modeling tool for a convenient representation of a joint probability distribution over a set of variables. BNs have been used to model uncertainty about events in many domains, such as machine diagnosis, medical diagnosis, risk analysis, and classification. A BN is an acyclic directed graph, in which nodes represent variables and edges represent direct dependencies between pairs of these variables. Each node is associated with a conditional probability distribution, which in the discrete case is represented by a conditional probability table (CPT). A BN model allows to calculate the posterior probability distribution over variables of interest given a set of observations, where an observation is an assignment of a value to a variable.

The posterior probability distributions over variables of interest change as we gather observations about a case at hand. Each new observation introduces information that usually makes the probability estimate more case-specific and, hence, more precise. A user applying the model may want to know, how future observations will impact the model's result. For example, a physician investigating a case of a patient with a chest pain may consider running some clinical tests after gathering information about patient's medical history and listening to patient's lungs. A question of much interest is whether the probability of pneumonia can go up or down and by how much as we obtain the results of the clinical tests. In other words, how will the posterior probability of pneumonia change when we feed the model with more observations about the patient case at hand.

One way of representing the uncertainty about a calculated quantity (this is, in case of a BN model, a posterior probability) is a confidence interval, which utilizes the probability distribution over the predicted value. Given that a BN is a complete specification of the joint probability distribution over its variables, we have all the necessary information to derive such intervals.

Most of the literature on uncertainty in results of Bayesian network inference focuses on the impact of possible imprecision in parameters of the network. Such uncertainty can be captured by means of error bars or uncertainty intervals (e.g., work by Donald and Mengersen (2014) or Van Allen et al. (2008)). If the imprecision in parameters can be expressed by intervals, it can be propagated over the model to derive uncertainty intervals over results (Fagiuoli and Zaffalon, 1998; Cano et al., 1993). Uncertainty over results has also been a focus of sensitivity analysis, which amounts to studying the impact of small changes in individual model parameters on the result. For example, Laskey (1995) describes the derivation of error bars for probability assessment. Even though the question posed in this paper is useful and asked by users of probabilistic decision support systems, we have not found any literature analyzing the uncertainty intervals for posterior probabilities in anticipation of future observations.

In this paper, we present a method for deriving uncertainty (variation) intervals over posterior probabilities due to unknown observations about the case. The starting point for our work is a BN model, and we assume that both its structure and its parameters are correct. Because the distribution over possible values of posterior probabilities given different observations is not necessarily parametric, we propose to use an empirical distribution. The number of possible combinations of observations is typically too large to analyze. In such situation, we simulate the observations by means of a stochastic sampling method based on posterior probability distributions over unobserved variables.

The remainder of this paper is structured as follows. Section 1 introduces notation and necessary definitions. Section 2 describes two simulation methods for deriving the variation intervals over posterior probabilities. Section 3 demonstrates and compares these methods. Section 4 concludes the paper with final remarks and discussion.

1. Definitions and notation

Throughout this paper, we will use capital letters, e.g., X, to denote random variables. Even though random variables in BNs may be continuous, we focus on variables with finite numbers of outcomes. Let $Val(X) = \{x_1, \ldots, x_{n_i}\}$ be a set of possible outcomes of a random variable X. An observation of a variable X is an assignment of one of its possible outcomes $X = x_j$, which we will shorten to x_j.

Let $\mathcal{G}(\mathbf{V}, \mathbf{E})$ be an acyclic directed graph, where $\mathbf{V}$ is a set of vertices (nodes) and $\mathbf{E}$ is a set of pairs (V, W) representing directed edges between nodes $V, W \in \mathbf{V}$. Let $\mathrm{Pa}(V)$ be a set of nodes that are immediate predecessors (parents) of V. Let $\mathrm{Ch}(V)$ be a set of vertices that are immediate successors (children) of V.

1.1 Bayesian network

A discrete Bayesian network (BN) is a pair $(\mathcal{G}, \Theta)$, where $\mathcal{G}(\mathbf{V}, \mathbf{E})$ consists of

- $\mathbf{V} = \{V_1, V_2, \ldots, V_n\}$ represents a set of random variables, each with a finite set of mutually exclusive states $Val(V_i)$ and
- a set of edges $\mathbf{E}$ that jointly model the independencies among variables $\mathbf{V}$;

Θ is a set of parameters $\{\theta_{v_{i,j}|c_k},\ v_{i,j} \in Val(V_i) \wedge c_k \in Val(\mathrm{Pa}(V_i))\}$, which define conditional probability distributions $\Pr(V_i|\mathrm{Pa}(V_i))$ for each V_i.

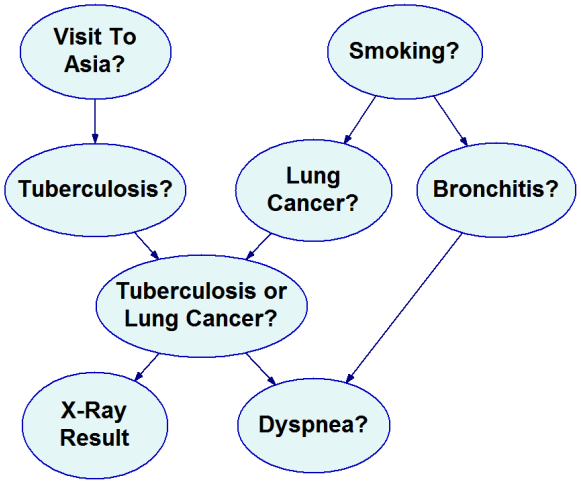


Figure 1: The ASIA Bayesian network (Lauritzen and Spiegelhalter, 1988).

Parameters $\theta_{v_{i,\bullet}|\bullet}$ of the conditional probability distribution of a variable V_i can be organized in a conditional probability table (CPT) that describes the probability distribution over V_i for all combinations of assignments to $\mathrm{Pa}(V_i)$. Figure 1 shows the ASIA model (Lauritzen and Spiegelhalter, 1988), which models the situation of a patient appearing in a clinic with dyspnea (shortness of breath). It consists of eight discrete random variables representing disorders (*Tuberculosis*, *Lung Cancer*, *Bronchitis*), historical data (*Visit to Asia*, *Smoking*), auxiliary variables (*Tuberculosis or Lung Cancer*), symptoms (*Dyspnea*) and examinations that a physician can perform (*X-Ray Result*).

1.2 Targets, evidence, and Markov blankets

Let $\mathbf{T} \subset \mathbf{V}$ be a set of variables of interest (targets). Let $\mathbf{S} \subset \mathbf{V}$ be all observable phenomena modeled by the BN, e.g., symptoms or patient history data in a medical decision support system. An evidence set $\mathbf{E}$ is a set of observations (assignments) ($\{v_{i_1,j_1}, \ldots, v_{i_k,j_k}\}$, where $\{V_{i_1}, \ldots, V_{i_k}\} = \mathbf{S}_O \subset \mathbf{S}$ is a set of variables with assignments in $\mathbf{E}$). A scenario $\mathbf{E}^* \supset \mathbf{E}$ is an evidence set that assigns outcomes to all variables in $\mathbf{S}$. We will denote by $\mathbf{S}_U$ the set of variables without associated assignment in $\mathbf{E}$ i.e., $\mathbf{S}_U = \mathbf{S} \setminus \mathbf{S}_O$. For example, in the ASIA model, variables *Tuberculosis*, *Lung Cancer* and *Bronchitis* compose the set of target variables $\mathbf{T}$. Variables *Visit to Asia*, *X-Ray Result*, *Dyspnea* and *Smoking* belong to the set $\mathbf{S}$ of observable phenomena. If we consider a patient with dyspnea, we have an evidence set consisting of one assignment $\mathbf{E} = \{dyspnea = present\}$. Based on this evidence set $\mathbf{E}$, we can calculate the posterior probability of the patient having tuberculosis $\Pr(Tuberculosis = present|\mathbf{E})$. Usually the term *probabilistic inference* refers to calculations of posterior probabilities (Lauritzen and Spiegelhalter, 1988). While the method proposed in this paper is general enough to apply to any calculated posterior probability in a Bayesian network, we focus on the posterior marginal probabilities of single outcomes.

The Markov blanket of a variable $V_i \in \mathbf{V}$ is the set $\mathbf{M}(V_i) \subset \mathbf{V}$ consisting of variables that are parents $\mathrm{Pa}(V_i)$, children $\mathrm{Ch}(V_i)$, and parents of its children $\mathrm{Pa}(\mathrm{Ch}(V_i))$, i.e.,

$$\mathbf{M}(V_i) = \mathrm{Pa}(V_i) \cup \mathrm{Ch}(V_i) \cup \mathrm{Pa}(\mathrm{Ch}(V_i)) \,.$$

$\mathbf{M}(V_i)$ represents all variables such that, when observed, make V_i independent of the remainder of the variables in the network. For example, in Figure 1, $\mathbf{M}(Smoking) = \{Lung\ Cancer, Bronchitis\}$, as variables *Lung Cancer* and *Bronchitis* make *Smoking* independent of the rest of the network.

We can extend the definition of Markov blanket to sets of variables $\mathbf{A} \subset \mathbf{V}$. $\mathbf{M}(\mathbf{A})$ is a union of Markov blankets $\mathbf{M}(V_i)$ of each variable $V_i \in \mathbf{A}$ excluding V_i, i.e.,

$$\mathbf{M}(\mathbf{A}) = \left(\bigcup_{V_i \in \mathbf{A}} \mathbf{M}(V_i) \right) \setminus \mathbf{A} .$$

If a Markov blanket $\mathbf{M}(V_i)$ contains a variable V_j, that is not observable (i.e., $V_j \in \mathbf{V} \setminus \mathbf{S}$), V_j cannot be used to screen V_i from the rest of the network. We will extend the definition of Markov blanket $\mathbf{M}(V_i)$ to an extended Markov blanket $\mathbf{M}^*(V_i)$, which we define as a set of observable variables that makes V_i independent from all the other observable variables. $\mathbf{M}^*(V_i)$ can be calculated recursively in the following way. We start with a set $\mathbf{C} = \{V_i\}$. We add all non-observable variables $V_j \in \mathbf{M}(\mathbf{C}) \cap (\mathbf{V} \setminus \mathbf{S})$ to $\mathbf{C}$. We repeat this procedure as long as $\mathbf{M}(\mathbf{C}) \cap (\mathbf{V} \setminus \mathbf{S}) \neq \emptyset$, in which case $\mathbf{M}^*(V_i) = \mathbf{M}(\mathbf{C})$.

1.3 Variation intervals over future probabilities

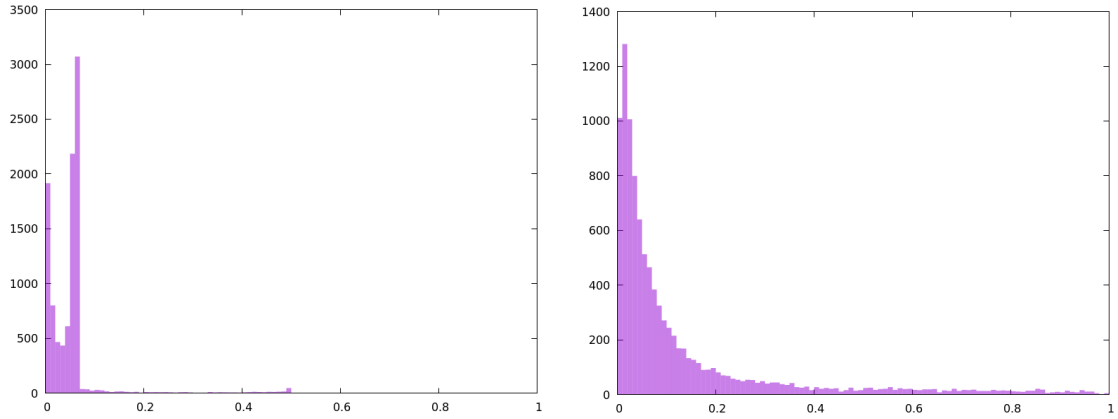
We are interested in anticipated changes in the posterior probability of a target variable due to possible future observations consistent with the evidence $\mathbf{E}$ at hand. Determining all possible future observations would require analyzing all possible scenarios $\mathbf{E}^* \supset \mathbf{E}$. Analyzing all these scenarios for a large model may be daunting. For example, the HEPAR II model[1] for supporting diagnosis of liver disorders (Oniśko et al., 2001) consists of 70 variables of which 61 are observable. The size of the complete set of scenarios for HEPAR II is over 3.78215×10^{21}.

In such a case, we can derive a sample of scenarios as described below. For a given evidence set $\mathbf{E}$, we obtain possible future observations by stochastic simulation, i.e., we draw outcomes from the posterior probability distribution of each observable variable in $\mathbf{S}$ to obtain a possible scenario of observations $\mathbf{E}^*$. We can repeat the simulation to get a sample of possible scenarios $\{\mathbf{E}^*_1, \ldots, \mathbf{E}^*_s, \ldots, \mathbf{E}^*_N\}$. If we calculate the posterior probabilities of an outcome of a target variable given each scenario (e.g., $\Pr(Bronchitis = present|\mathbf{E}^*_s)$), we will obtain a sample of possible future probabilities of that outcome.

Figure 2 shows two histograms of posterior probability of assignments to two target variables in the HEPAR II model, $\Pr(Carcinoma = present|\mathbf{E}^*)$ (a) and $\Pr(Chronic\,Hepatitis = active|\mathbf{E}^*)$ (b). Both histograms were generated by sampling (as described above) with the evidence set $\mathbf{E} = \{Hepatitis\,B\,Antigen = absent\}$.

Histograms such as those pictured in Figure 2 show typically a wide spread. For example, the values in the histogram (b) cover the entire range $(0, 1)$. It seems that reporting the range of possible values is, therefore, quite useless. Because both histograms show some central tendency, a trimmed range (for example, one showing 95% of all values) will be more informative. To this effect, we can trim the extreme 2.5% of sampled values at each end. The precise cut-off points can be interpreted as a numerical estimate of the 95% confidence interval over the current value of the target probability calculated by the model in the light of future observations.

1. Available through several public Bayesian network repositories.



(a) $\Pr(Carcinoma = present|\mathbf{E}^*)$ (b) $\Pr(Chronic\,Hepatitis = active|\mathbf{E}^*)$

Figure 2: Histograms representing samples of posterior probabilities values given one assignment to a variable in HEPAR II model.

2. Calculation of the variation interval over future posterior probabilities

In this section, we formalize the procedure described in Section 1.3 by proposing two methods for sampling the possible posterior probabilities in anticipation of future observations. The first method (Algorithm 1) is based on an exhaustive instantiaton of all observable variables. We follow this by an improved approach (Algorithm 2) that narrows down the number of sampled variables to the extended Markov blanket of the target variable.

Algorithm 1 iterates through the set of all observable variables to assign a value to each unobserved variable (line 4). To draw an outcome for a variable, it calculates the posterior probability distribution over its outcomes given evidence (line 5). Then, it samples an outcome from the calculated posterior probability distribution (line 6). Having outcomes assigned to all the observable variables, the algorithm calculates the posterior probability of the pursued outcome of the target variable, which amounts to one sample (lines 9-10). Based on the sample, we derive a variation interval (empirical confidence interval) over the posterior probability of the pursued outcome (line 12).

Each calculation of the marginal posterior probability distribution of a variable involves a call to a Bayesian network inference algorithm. Each derivation of the variation interval involves $O(N \times (|\mathbf{S}| - |\mathbf{E}|))$ calls of the inference algorithm, where N describes the number of samples, $|\mathbf{S}|$ is the number of observable variables, and $|\mathbf{E}|$ is the number of observations. Probabilistic inference is worst-case NP-hard (Cooper, 1990) and even with the fastest algorithm available may turn out to be too slow for interactive systems.

Generation of samples in Algorithm 1 can be improved by exploring independence between the target variable and other variables conditional on the target variable's Markov blanket. Because in practice not all model variables are observable, we use the concept of the extended Markov blanket, introduced in Section 1.2. Extended Markov blanket screens off the target variable given a minimal set of those variables that are observable. This mitigates the problem of multiple calls to Bayesian network inference algorithm by reducing the set of sampled variables to those in the extended Markov blanket of the target variable.

CISampleAllObservable

Input : BN $(\mathcal{G}, \Theta)$, target variable V_t, target assignment $v_{t,j}$, evidence $\mathbf{E}$, unobserved variables $\mathbf{S}_U$, number of samples N, confidence level $1 - \alpha$

Output: Sample H of possible probabilities $\Pr(v_{t,j}|\mathbf{E}^*)$, variation interval (p_L, p_U)

1 $H \leftarrow \emptyset$
2 **for** $k = 1, \ldots, N$ **do**
3 $\quad \mathbf{E}^* \leftarrow \mathbf{E}$
4 $\quad$ **foreach** $V_i \in \mathbf{S}_U$ **do**
5 $\quad\quad$ Calculate $\Pr(V_i|\mathbf{E}^*)$
6 $\quad\quad$ Draw $v_{i,k} \sim \Pr(V_i|\mathbf{E}^*)$
7 $\quad\quad$ $\mathbf{E}^* \leftarrow \mathbf{E}^* \cup \{v_{i,k}\}$
8 $\quad$ **end**
9 $\quad$ Calculate $\Pr(V_t|\mathbf{E}^*)$
10 $\quad$ $H \leftarrow (H, \Pr(v_{t,j}|\mathbf{E}^*))$
11 **end**
12 Construct $1 - \alpha$ variation interval (p_L, p_U) using sample H

Algorithm 1: The algorithm for deriving the variation interval for posterior probability values by sampling the space of assignments of all unobserved variables.

Algorithm 2 starts with determining the extended Markov blanket of the target variable (lines 1-10). In particular, we create two sets to store unprocessed ($\mathbf{A}$) and processed ($\mathbf{A}_D$) non-observable variables. After initialization (lines 1-3), we are recursively collecting variables from Markov blanket $\mathbf{M}(V_i)$ (lines 8-9) of a variable $V_i \in \mathbf{A}$ and moving V_i to the set $\mathbf{A}_D$ (lines 6-7). The remainder of the algorithm (lines 11-22) is similar to Algorithm 1, except for line 14, where we replaced $\mathbf{S}_U$ by $\mathbf{M}^*(V_t) \setminus \mathbf{S}_O$. As a result, Algorithm 2 involves $O(N \times (|\mathbf{M}^*(V_t) \setminus \mathbf{S}_O|))$ calls to the inference algorithm.

3. Evaluation of the proposed method

We applied our algorithms for calculating the 95% variation intervals over the posterior marginal probability of a target outcome to three practical Bayesian network models described below.

HEPAR II is a Bayesian network model for diagnosis of liver disorders (Oniśko et al., 2001), available from several public Bayesian network repositories. HEPAR II consists of 70 variables, arranged in three groups: patient history and risk factors (18 variables), diseases (9 target variables), and symptoms or test results (43 variables). HEPAR II's graph models the causal structure of the domain. For our tests, we picked various target variables from among the nine disease variables.

MORTALITY90D is a Bayesian network model for forecasting mortality of patients 90 days after heart transplant (Kanwar et al., 2017). The structure of MORTALITY90D follows a Tree-augmented Naive Bayes (TAN) model with one class variable representing *mortality* and 27 predictor variables. The TAN structure forces two types of edges: connecting *mortality* with all predictor variables and those forming a tree structure among all predictor variables. The Markov blanket of *mortality* consists of all predictor variables.

CPCS179 is a Bayesian network model created from the knowledge base of the Computer-based Patient Case Simulation (CPCS) system (Pradhan et al., 1994). CPCS179 consists of 179

CISampleExtendedMarkovBlanket

Input : BN $(\mathcal{G}, \Theta)$, target variable V_t, target assignment $v_{t,j}$, evidence $\mathbf{E}$, observable variables $\mathbf{S}$, number of samples N, confidence level $1 - \alpha$

Output: Sample H of possible probabilities $\Pr(v_{t,j}|\mathbf{E}^*)$, variation interval (p_L, p_U)

$\mathbf{M}^*(V_t) \leftarrow \mathbf{M}(V_t) \cap \mathbf{S}$
$\mathbf{A} \leftarrow \mathbf{M}(V_t) \setminus \mathbf{S}$
$\mathbf{A}_D \leftarrow \emptyset$
while $\mathbf{A} \neq \emptyset$ **do**
    pick any V_i from $\mathbf{A}$
    $\mathbf{A} \leftarrow \mathbf{A} \setminus \{V_i\}$
    $\mathbf{A}_D \leftarrow \mathbf{A}_D \cup \{V_i\}$
    $\mathbf{A} \leftarrow \mathbf{A} \cup (\mathbf{M}(V_i) \setminus (\mathbf{S} \cup \mathbf{A}_D))$
    $\mathbf{M}^*(V_t) \leftarrow \mathbf{M}^*(V_t) \cup (\mathbf{M}(V_i) \cap \mathbf{S})$
end
$H \leftarrow \emptyset$
for $k = 1, \ldots, N$ **do**
    $\mathbf{E}^* \leftarrow \mathbf{E}$
    foreach $V_i \in \mathbf{M}^*(V_t) \setminus \mathbf{S}_O$ **do**
        Calculate $\Pr(V_i|\mathbf{E}^*)$
        Draw $v_{i,k} \sim \Pr(V_i|\mathbf{E}^*)$
        $\mathbf{E}^* \leftarrow \mathbf{E}^* \cup \{v_{i,k}\}$
    end
    Calculate $\Pr(V_t|\mathbf{E}^*)$
    $H \leftarrow (H, \Pr(v_{t,j}|\mathbf{E}^*))$
end
Construct $1 - \alpha$ variation interval (p_L, p_U) using sample H

Algorithm 2: The Algorithm for deriving the variation interval for posterior probability values by instantiating variables of the extended Markov blanket of the target variable.

variables connected by 239 edges and, similarly to HEPAR II, its graph follows the causal structure of the domain. We treat this model as an example of a sizable Bayesian network. We chose the following two variables as targets for our tests: *Alcoholic Hepatitis*, with one parent variable and 26 children variables, and *Cholestasis*, with one parent variable and 14 children variables. We treated the remaining variables as observable.

3.1 Examples of the derived variation intervals

To demonstrate the usefulness and practical behavior of the variation intervals over future observations, we performed several simulations of a diagnostic process using the HEPAR II model (we used a handful of real patient cases from a data set used for learning the parameters of the HEPAR II model). For each target variable V_t and an evidence set $\mathbf{E}_i$, we followed the following procedure:

1. From the set of unobserved variables, choose the variable that carries the most information measured by cross-entropy for target V_t given already observed values. This gave us a realistic order of observations during the diagnostic process: from the most to the least informative evidence.

2. Enter the observation from the evidence set $\mathbf{E}_i$ for the chosen variable into the model.

3. Calculate the posterior marginal probability distributions of the target variables.

4. Derive variation intervals for those probabilities.

5. Repeat all these steps until all observations belonging to evidence set $\mathbf{E}_i$ have been made.

Figure 3 shows eight examples of 95% variation intervals over the posterior probability of *Chronic Hepatitis* being persistent (a), *Chronic Hepatitis* being active for two different cases (b-c), *PBC* (primary biliary cirrhosis) (d) being present, *Toxic Hepatitis* being present (d), *Cirrhosis* being compensated for three different cases (f-h). There are 61 possible observations (referring to risk factors, symptoms, and test results in the HEPAR II model) for each case and they are made individually from left to right. We used a fixed number of $N = 1,000$ samples in each experiment. The solid line running from left to right demonstrates the development of the probability of the target event in question as new observations are made. The area around the probability line shows the variation interval over the probability at each point in time. Please note that the variation intervals start by being very wide in the beginning, which corresponds to the situation when nothing about the patient is known. As more and more evidence is accumulated, the variation intervals narrow, to the point of becoming either a point probability (when all possible 61 observations have been made) or a fixed interval, when some of the observations have never been made in a patient's case.

3.2 Computation time

To compare the computation time of the two proposed algorithms, for each of the three models we generated 100 test records containing values of the observable variables. We used a version of probabilistic logic sampling (Henrion, 1988), making sure that 50% of all values are missing at random. For each record in the generated data sets, we derived 95% variation interval of posterior probability of one target variable (randomly chosen among targets in the model), using both Algorithm 1 and Algorithm 2. We ran our tests on a computer with Intel® Core™ i5-5200U CPU @ 2.20GHz

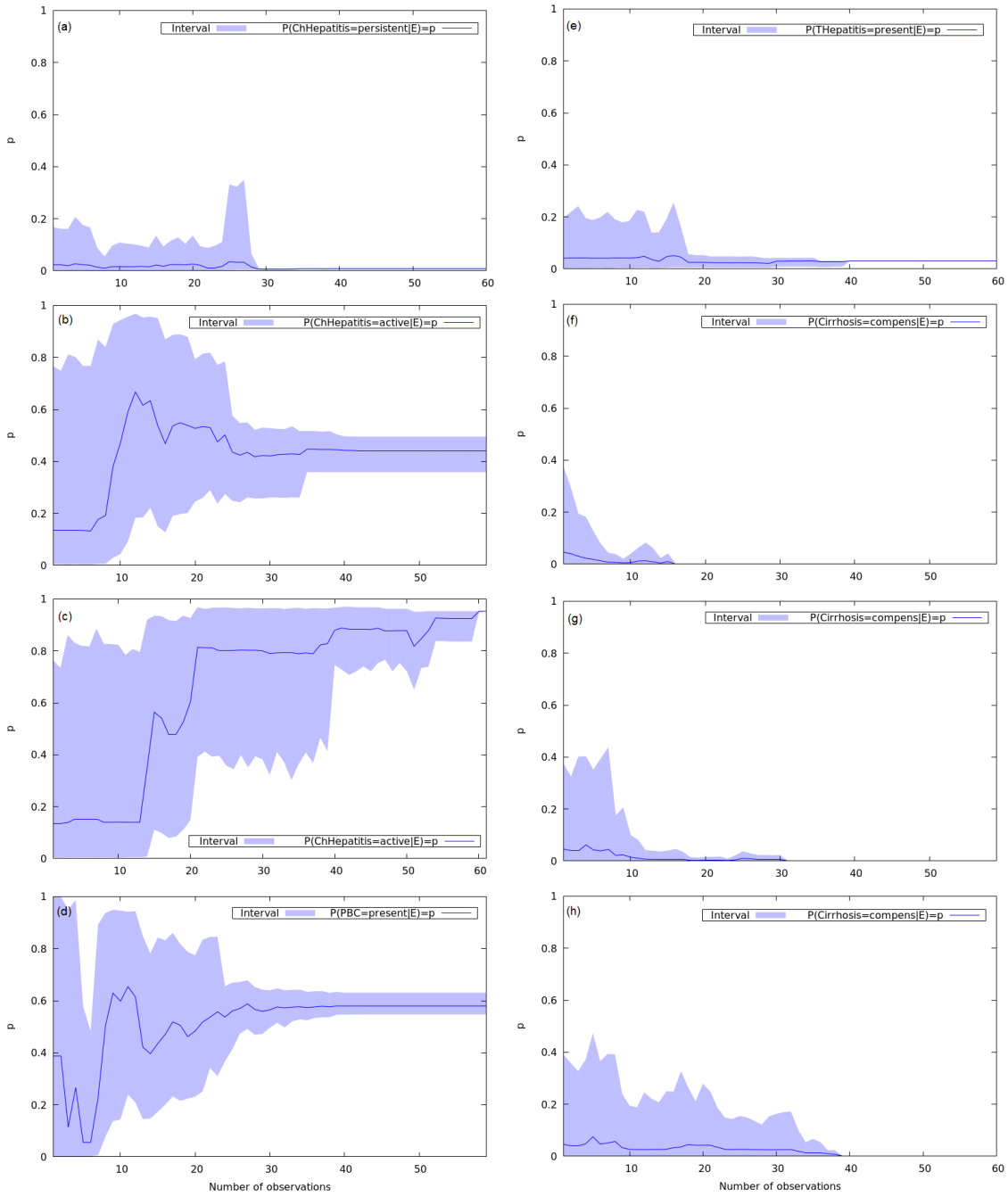


Figure 3: Examples of 95% variation intervals over the posterior probability of *Chronic Hepatitis* being persistent (a), *Chronic Hepatitis* being active for two different cases (b-c), *PBC* (primary biliary cirrhosis) being present (d), *Toxic Hepatitis* being present (d), *Cirrhosis* being compensated for three different cases (f-h) in the HEPAR II model.

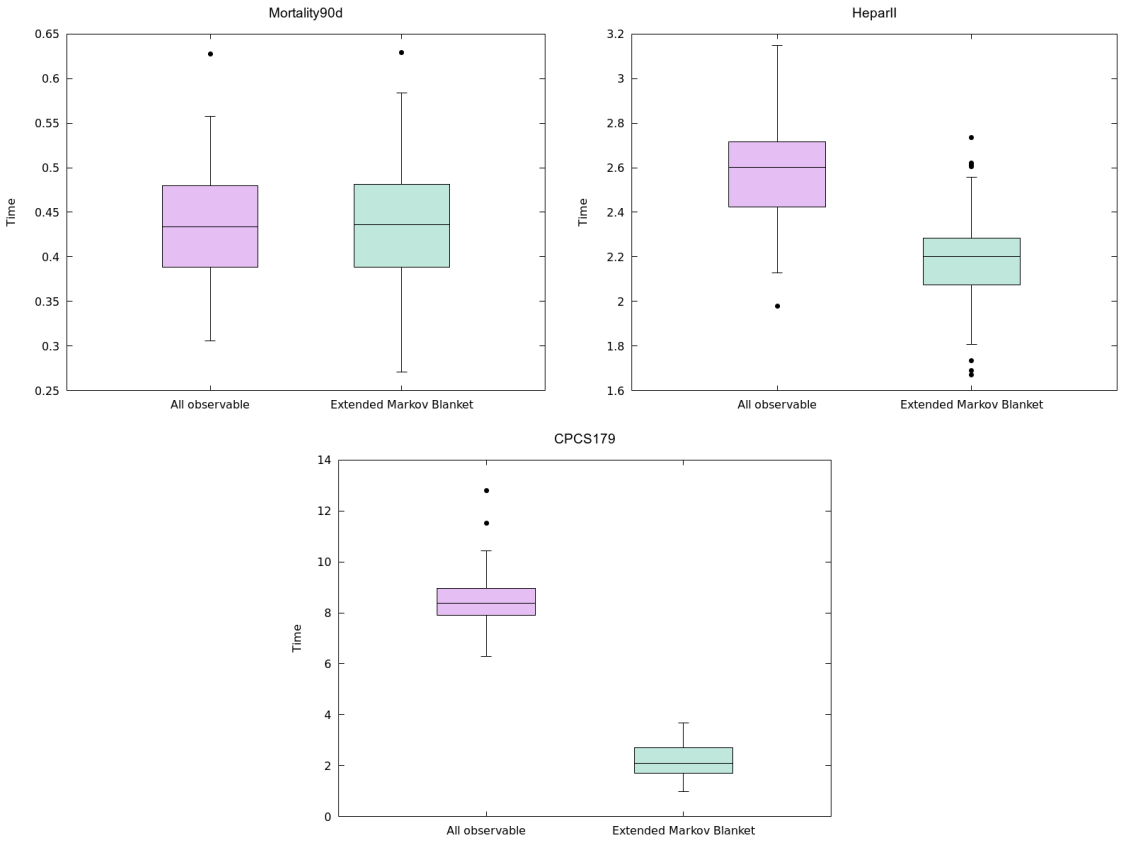


Figure 4: Box plots comparing computation times of variation intervals for posterior probabilities with both versions of the algorithm (measured in seconds): the Algorithm 1 (sampling all observable variables) and the Algorithm 2 (sampling variables from extended Markov blanket).

processor, 8GB memory, 32KB/256KB/3MB processor cache, running Ubuntu Linux 16.04.1 LTS x86-64 distribution. Our implementation used SMILE (Druzdzel, 1999) Bayesian network software library.

Figure 4 shows box plots representing time spent by each of the algorithms. For the MORTALITY90D model (tree augmented naive Bayes), derivation of variation intervals takes similar amount of time. This is understandable given that the Markov blanket of the target variable in a TAN model consists of all remaining variables and Algorithm 2 practically deteriorates into Algorithm 1. The slight difference between whiskers results from a different order of variables in the simulation process. For both, the HEPAR II and CPCS179 models, Algorithm 2 is much faster ($p < 10^{-57}$ for HEPAR II model and $p < 10^{-115}$ for CPCS179 model), as it takes advantage of the extended Markov blankets of the target variables. In all three cases, the absolute computation time seems acceptable from the point of view of an interactive user interface.

4. Discussion and future work

In this paper, we proposed calculating variation intervals over posterior probabilities in Bayesian networks in anticipation of future observations. We proposed a simple algorithm for deriving such variation intervals and an improvement on this algorithm based on the concept of an extended Markov blanket, which reduces the amount of computation needed to derive the variation intervals. We presented examples of variation intervals calculated for practical Bayesian network models and showed that the intervals change as expected when more information becomes available. If used in decision support systems, variation intervals over future posterior probabilities seem to provide interesting insight in the change of the system's output.

We employed the variation intervals in a practical decision support system incorporating the MORTALITY90D model in a medical decision support system calculating mortality risk of patients after a heart transplant. Its users found the plots of 95% variation intervals over the posterior probabilities highly insightful in the process of diagnosis and were delighted with the new feature. Although, for other models, it may be necessary to use much wider intervals.

Our implementation uses a fixed number of $1,000$ samples to derive the example variation intervals in Sections 3.1 and 3.2. This number is sufficient to make the variation intervals statistically reliable. We believe that it may be possible to reduce this number and determine the necessary sample size to obtain reasonable precision of interval bounds. This should further improve the efficiency of the method.

Acknowledgments

Partial support for this work was provided by National Institute of Health grants U01HL101066-01 and 1R01HL134673-01 and a Department of Defense grant W81XWH-17-1-0556. The CORA and PHORA projects have been an inspiration for our work and we thank the team members for a stimulating environment to work in. Jim Antaki and Lisa Carey-Lohmueller, in particular, persistently, although patiently asked us for variation intervals over posterior probabilities. We thank Agnieszka Druzdzel for providing insight into the HEPAR II model. We used a free academic license of SMILE and GeNIe, available through BayesFusion, LLC's website (http://www.bayesfusion.com/).

References

J. Cano, M. Delgado, and S. Moral. An axiomatic framework for propagating uncertainty in directed acyclic networks. *International Journal of Approximate Reasoning*, 8(4):253–280, 1993.

G. F. Cooper. The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence*, 42:393–405, 1990.

M. R. Donald and K. L. Mengersen. Methods for constructing uncertainty intervals for queries of Bayesian nets. *Australian & New Zealand Journal of Statistics*, 56(4):407–427, 2014. ISSN 1467-842X.

M. J. Druzdzel. SMILE: Structural Modeling, Inference, and Learning Engine and GeNIe: A development environment for graphical decision-theoretic models. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI–99) and the eleventh Innovative Applica-*

tions of Artificial Intelligence Conference (IAAI-99), pages 902–903, Menlo Park, CA, July 18–22 1999. AAAI Press/The MIT Press. ISBN 0–262–51106–1.

E. Fagiuoli and M. Zaffalon. 2U: An exact interval propagation algorithm for polytrees with binary variables. *Artificial Intelligence*, 106(1):77–107, 1998.

M. Henrion. Propagation of uncertainty by probabilistic logic sampling in Bayes' networks. In *Uncertainty in Artificial Intelligence*, volume 2, pages 149–164, 1988.

M. K. Kanwar, L. C. Lohmueller, R. L. Kormos, N. A. Loghmanpour, R. L. Benza, R. J. Mentz, S. H. Bailey, S. Murali, and J. F. Antaki. Low accuracy of the HeartMate risk score for predicting mortality using the INTERMACS registry data. *ASAIO Journal*, 63(3):251–256, 2017.

K. B. Laskey. Sensitivity analysis for probability assessments in Bayesian networks. *IEEE Transactions on Systems, Man, and Cybernetics*, 25(6):901–909, 1995.

S. L. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 157–224, 1988.

A. Oniśko, M. J. Druzdzel, and H. Wasyluk. Learning Bayesian network parameters from small data sets: Application of Noisy-OR gates. *International Journal of Approximate Reasoning*, 27 (2):165–182, 2001.

J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

M. Pradhan, G. Provan, B. Middleton, and M. Henrion. Knowledge engineering for large belief networks. In *Proceedings of the Tenth International Conference on Uncertainty in Artificial Intelligence*, pages 484–490. Morgan Kaufmann Publishers Inc., 1994.

T. Van Allen, A. Singh, R. Greiner, and P. Hooper. Quantifying the uncertainty of a belief net response: Bayesian error-bars for belief net inference. *Artificial Intelligence*, 172(4-5):483–513, 2008.

What can the PGM community contribute to the 'Bayesian Brain' hypothesis?

Johan Kwisthout J.KWISTHOUT@DONDERS.RU.NL
Donders Institute for Brain, Cognition and Behaviour
Radboud University Nijmegen
PO Box 9104, 6500HE Nijmegen, The Netherlands

Abstract

Despite the now common view amongst neuroscientists that the brain effectively approximates Bayesian inferences (known as the 'Bayesian Brain hypothesis'), there are only few researchers in the PGM community currently working in this research area. We believe that this is partially due to a misunderstanding of the theoretical challenges that theoretical neuroscience currently faces and the potential contribution that the PGM community can offer in interdisciplinary research. With this paper we hope to remedy such misunderstandings and invite the community to contribute to the mutual benefit of neuroscience and AI alike.

Keywords: Bayesian Brain hypothesis; neuroscience; interdisciplinary research.

1. Introduction

When discussing recent advances in neuroscience–that postulate that the human brain is at its essence just an approximate Bayesian inferential machine–with scholars in the Probabilistic Graphical Models (PGM) community, our research group occasionally receives lukewarm responses that can best be paraphrased as "I'm just not interested in the brain as an application area of my research". Although there are few things as personal as a research agenda, we still feel that this lack of interest may be at least partially due to a) a misconception of the questions that are currently being addressed in neuroscience and b) lacking some 'insiders insight' in the contribution that the PGM community can offer in interdisciplinary research. With this paper we hope to remedy both.

Our approach here is orthogonal and complementary to the approach put forward by Bielza and Larrañaga (2014) who described the use of Bayesian networks as *tools for* neuroscientific research, such as reconstructing human brain activity from fMRI data (Schoenmakers et al., 2015), spatial component analysis for Alzheimer's disease diagnosis (Illan et al., 2014), or classification of interneurons (Mihaljević et al., 2014). This is an important area of PGM research, but already sufficiently covered in Bielza and Larrañaga's special issue (Bielza and Larrañaga, 2014). In contrast, in our approach we are interested in (computations on) graphical models as *objects of study in* neuroscience, i.e., computational-level explanations of the brain's information processing activity.

We will give a short overview of the increasingly popular 'Bayesian Brain' hypothesis in neuroscience, in particular its 'predictive processing' manifestation. We will then identify three concrete research areas within this topic where contributions from the PGM community can actually have a huge scientific impact. After identifying some potential pitfalls in such interdisciplinary research, including a discussion of the specific (and sometime peculiar) connotations of the neuroscience community with respect to concepts like 'Bayesian,' 'uncertainty,' and 'prior,' we will conclude with an invitation to the community to contribute.

2. The Brain as 'Application Area' for PGM

Herman von Helmholtz (1867) is traditionally seen as the originator of the view of human perception as (statistical) inference to the best explanation of the causes of the perceptual input. The suggestion that the human brain can be seen as performing some approximate Bayesian inference (integrating prior expectations with newly arriving information) was coined as early as 1957 by Edwin T. Jaynes (first published by Jaynes (1988)). Peter Dayan and colleagues further explored these ideas and proposed the notion of the *Bayesian Brain* (Yu and Dayan, 2005), emphasizing on the basis of psychophysical evidence that human perception actually is 'Bayes' optimal' in combining priors and new signals. The *Bayesian coding* hypothesis (Knill and Pouget, 2004) postulates that the brain indeed represents probability distributions in populations of neurons.

In recent years, the *Bayesian Brain* hypothesis has become increasingly popular due to the emergence of Karl Friston's *free energy principle*, providing for a biological and physical foundation; the *predictive processing* view of the brain as a 'prediction machine' that minimizes computational effort by trying to predict its inputs, and the *spiking neural network* research area that shows that probability distributions can be encoded and sampled from using power-efficient networks of spiking neurons. We will elaborate more on these three important recent developments.

2.1 The free energy principle

Friston's *free energy principle* (Friston, 2009, 2010) postulates that any biological system that 'resists a tendency to disorder' – be it a single cell or a social network – effectively aims to minimize free energy. In thermodynamics, free energy is the amount of energy that is potentially available, but not put to effective use. In information theory, it is a measure on the discrepancy between our observation of the world and our model of the world, which becomes manifest as the *prediction error* between predicted and observed world state. A biological system that aims to defy disorder seeks to lower expected entropy (the average of surprise of future outcomes). It can do so by minimizing prediction error, that is, aiming to make the predicted world state match the observed world state (adapting one's models of the world), or vice versa (changing one's sensory input by acting upon the world). Because biological systems must remain within certain boundaries to exist, their models of what the world should look like (e.g., have access to a sufficient, but not excess, amount of oxygen to maintain homeostasis) and how they currently perceive the world (e.g., shortage of oxygen) should match, and if not, actions are taken to minimize this prediction error (e.g., breathe faster and deeper). Friston (2009, p.295) summarizes this by postulating that *(i) agents resist a natural tendency to disorder by minimizing a free-energy bound on surprise; (ii) this entails acting on the environment to avoid surprises, which (iii) rests on making Bayesian inferences about the world.*

2.2 Predictive processing

The Predictive Processing account proposes that the brain continuously predicts its inputs in a hierarchical cascade of (increasingly more concrete) probabilistic predictions (Clark, 2013, 2015; Hohwy, 2013). For example, when observing a bowler on a bowling lane, contextual information ("this bowler already hit three strikes in this game") will generate predictions for the result of the throw ("many pins will fall down"). Based on that expectation, more specific predictions will be made for the throwing kinematics, the ball trajectory, where the ball will hit the pins, etc. Ultimately

this will generate predictions for sensory inputs to, e.g., the retina. Violations of predictions (a miss) will yield prediction errors that need to be 'explained away' by updating ones hypotheses ("even good bowlers will sometimes fail to throw a strike"), taking new contextual information into consideration ("the bowler seems to have injured his wrist whilst throwing") etc. Predictions are made with a specific precision, reflecting uncertainty about outcomes due to limited exposure (i.e, reducible uncertainty) or due to inherent stochasticity of the data-generating process (i.e., irreducible uncertainty). Prediction errors are used to update the generative models to minimize the reducible uncertainty.

The computations 'under the hood' of this conceptual description can be described and analyzed as various computations on causal Bayesian networks, such as the computation of posterior probability distributions, updating hyperparameters of distributions, and tuning of selected parameters of the network (Kwisthout et al., 2017). Despite its popularity as a unifying theory, it is far from clear what the brain's approximation algorithms actually look like; in Clark's words: *What do the local approximations to Bayesian reasoning look like as we depart further and further from the safe shores of basic perception and motor control? What new forms of representation are then required, and how do they behave in the context of the hierarchical predictive coding regime (Clark, 2013, p.201)?*

2.3 Networks of spiking neurons

One of the most promising computational models of neuronal computation in general is the recurrent *network of spiking neurons* model (Maass, 2014). These biologically inspired networks mimic Boltzmann machines (neural networks that represent a probability distribution that can be sampled from), with a key difference that the neurons are not outputting a zero or one state, but a *spike*; a brief burst of energy. These networks are energy-efficient and stochastic in nature and they can represent, and reason with, arbitrary probability distributions by means of stochastic sampling in winner-take-all microcircuits (Buesing et al., 2011; Pecevski et al., 2011; Habenschuss et al., 2013). It has been proposed that such sampling methods (like MCMC sampling) are the most promising techniques to describe actual stochastic inferences in the brain (Tenenbaum et al., 2011). Because of their efficiency – the brain uses a mere 25W of energy – these networks are potentially crucial for future generations of computer hardware by utilizing (rather than trying to filter) the noise that is inherent at the nano-scale (Hamilton et al., 2014). No free lunch is offered, though: As approximate Bayesian inference is an intractable problem (Dagum and Luby, 1993; Kwisthout, 2018), there will be problem instances where the convergence time of the network will grow exponentially with the input size, in particular in networks with extreme probability distributions (Maass, 2014).

In terms of Marr's levels of explanation (Marr, 1982), one can see the free energy principle as aiming to answer the 'why' of the Bayesian Brain hypothesis, the predictive processing account describes 'what' is actually being computed, whereas the 'spiking neurons' community studies the algorithmic 'how' aspect of approximate Bayesian computations in the brain. Where the free energy/predictive processing and the networks of spiking neurons communities were traditionally relatively isolated – as a proxy, one could see them as exponents of the *UK*, respectively *Continental* approach towards theoretical neuroscience – there have been recent mutual research events (for example at the European Institute for Theoretical Neuroscience in Paris) that try to bridge the gap between both communities.

2.4 Organization of this paper

All these developments support the 'Bayesian' view of the brain as it is currently dominant in contemporary neuroscience. We believe that this opens up a significant area of research for the PGM community. In the remainder of this paper we will further elaborate on this. We will show how a formal and computational background can help to bring conceptual clarity and formal rigidity to the field; how neuroscience is in urgent need for new algorithms, implementations, and complexity analyses that computer scientists and AI practitioners can provide, and where new questions in the 'meta'-theory of learning and modifying Bayesian networks emerge.

3. Conceptual Clarity and Rigidity

An important area where researchers with a strong background in computational and formal modeling can make vital contributions is in offering conceptual clarity and formal rigidity, translating verbal theories into complete and consistent computational models, thus exposing ambiguities and gaps in the theory and explicating 'design choices' and their computational consequences (Otworowska et al., 2015). Examples are in the formal explication of the role and nature of the underlying principles of predictive processing (Phillips, 2017; Kay and Phillips, 2011; Thornton, 2017), critically assessing the validity of simplifying assumptions (Otworowska et al., 2014, see also Figure

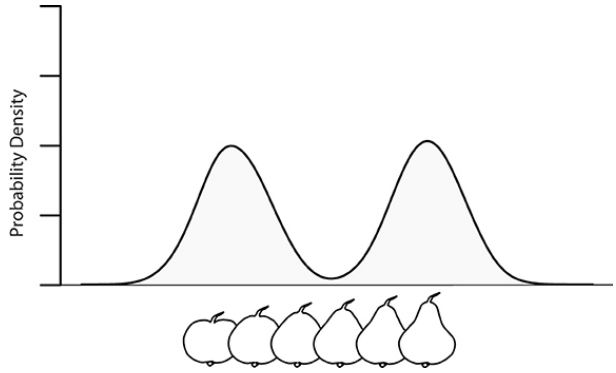


Figure 1: Recognition density for apples and pears based on the shape of the bulbous cone. Observe that, based on the frequency of apple-shapes, pear-shapes, and 'intermediate shapes' in the world, this recognition density cannot be assumed to be a simple Gaussian density; a violation of the Laplace assumption in Friston (2010). Picture reprinted (with permission) from Otworowska et al. (2014).

In the predictive processing theory, precision-weighted stochastic predictions are compared with actual observations and only the residual (non-predicted) signal is processed by the brain. Here, 'processed' means that by belief revision or by intervention the model and the reality are adapted to converge; a process denoted by prediction error minimization. For example, when we are tossing a coin to decide which team will start a match, initially we have uniform probability distributions predicting who wins the toss (the home or away team), what the outcome of the coin toss is (heads

or tails), and what visual stimuli we observe (either side of the coin). Note that these predictions are uncertain due to the inherent stochasticity of tossing coins, and will inevitably induce a prediction error when the coin lands as this will generate one bit of information that could not yet be predicted. This information is propagated 'upwards' by the prediction error minimization mechanism: the outcome is updated to 'heads,' which induces a prediction error with the original uniform prediction; in turn, the winner of the toss is updated to the away team to minimize this prediction error. Prediction error minimization is thus the mechanism by which information is processed in the brain.

Prediction errors, however, are dependent on the state space of the prediction and its granularity (the number of categories distinguished). In the absence of a coin we might have used a regular die and predict 'odd' or 'even' instead. We thus *lower* the typical state space of a die throw. Similarly, we might think of different sets of predicted inputs made by a couple strolling through the forest on a Sunday afternoon and an arborist looking for potentially hazardous situations in the same forest. From a modeling perspective: When we move from Gaussian densities to describe predictions in early vision or simple motor control to discrete probability distributions to describe higher cognitive capacities, we need to define what our categories are, and the granularity of our categories determines the prediction error. If we interpret the outcome of a die throw as odd or even, the prediction error decreases from 2.58 bits to 1 bit. This observation—made from an information-theoretic point of view—led to a further refinement of the predictive processing account with the notion of *levels of detail* of models and predictions (Figure

4. Theory, Algorithms, and Analysis

Most, if not all, computational problems in Bayesian networks are intractable. For example, inference is PP-complete (Littman et al., 1998), which implies that there cannot exist efficient approximation algorithms in general, unless BPP equals PP; casting a possible shadow over the biological plausibility of the Bayesian brain hypothesis. It has been suggested (e.g., (Clark, 2013, p.25, p.31)) that processing only the prediction error is less computationally demanding as processing the entire input and that predictive processing thus allows for a tractable implementation of the Bayesian Brain hypothesis. This assumption, however, does not (by and of its own) render inferences tractable. It was shown that processing even a single bit of prediction error is an NP-hard problem (Kwisthout, 2014). Recent developments in the area of *fixed-parameter tractability* allow for the analysis of stochastic computations where the probability of answering incorrectly is parameterized, rather than the computation time (Kwisthout, 2015, 2018). This allows for the study of so-called *fixed error randomized tractable* approximations, relative to 'ecologically valid' parameters, viz. parameters that can plausibly be assumed to be small in the computations as performed by the brain. In a separate paper submitted to this conference we show that the (relative) size of the prediction error plays virtually no role at all in tractability considerations: approximations are intractable or tractable, relative to a set of parameters, irrespective of the size of the prediction error (Donselaar, 2018). This effectively defies Clark's assertion; the biological validity of constraining parameters that *do* render approximation tractable, such as the local variance bound (Dagum and Luby, 1997), is currently under investigation.

Apart from process-level considerations (under what constraints can the approximations postulated by predictive processing be tractable), one can study the properties and plausibility of neuronal implementations of such approximations using networks of spiking neurons. Crucial properties here are the power efficiency of such networks (Maass, 2014), the nature of the *noise* in the brain and its

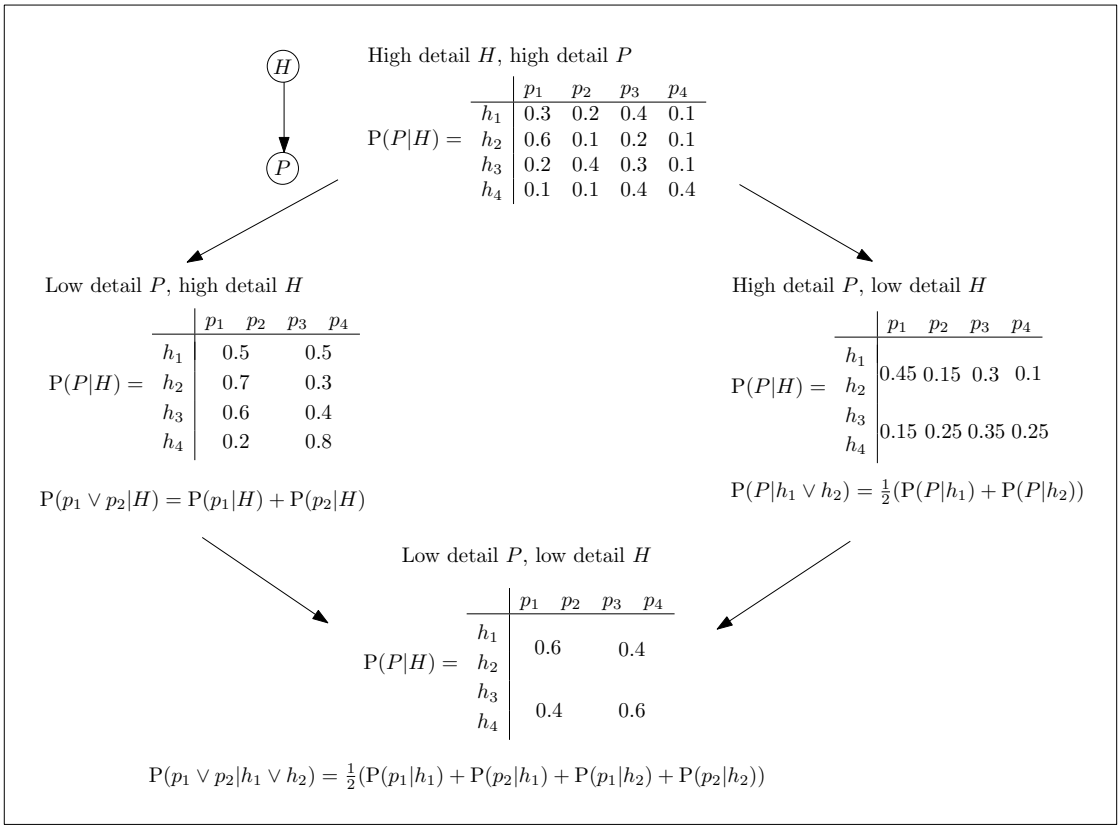


Figure 2: A formalization of the relationship between different levels of detail of hypotheses and predictions. Observe that actual hypotheses, as well as predictions, can be *clustered*, re-defining the conditional probability distributions in a straightforward way.

---

consequences for efficient sampling (Habenschuss et al., 2013), and the general question how many resources are needed for effective computations (Maass, 2000). Computational complexity theory offers an indication of the resources needed for a particular computational problem to be solved, as a function of the input size of a problem. These resources – most notably, time and memory – are typically fairly coarse and built on a theoretical abstract model of computation: Turing machines. Here, the 'time' resource refers to the number of state transitions in the machine, and the 'memory' resource refers to the number of memory cells on the tape that are used. It has been proposed by a working group at the Dagstuhl seminar on Resource-Bounded Problem Solving (seminar 14341) to have a more refined, brain-focused model of computation in the brain, based on networks of spiking neurons, and have complexity measures based on brain resources, such as spiking rates, network size, and connectivity (Haxhimusa et al., 2014). The development of such a model of computation would allow for seminal contributions to the Bayesian Brain hypothesis by analyzing the fundamental limits of brain computations.

5. Meta-theory of Bayesian Networks

When a prediction error is to be accounted for, one can either update ones current beliefs about the actual hypotheses, act upon the world in order to bring the reality closer to the desired state, or try to reduce uncertainty by observing hidden variables. These predictive processing sub-processes (belief revision, intervention, and adding observations) correspond to aspects of parameter tuning and sensitivity analysis (Coupé et al., 2000), counterfactual and prospective reasoning (Pearl, 2000), and selecting evidence (van der Gaag and Bodlaender, 2011). Several conceptual issues are still not resolved; for example, how counterfactual models can be built up and how we can use structure equation models to reason about *what* action we should undertake. Algorithmic and analytical aspects of these problems are of direct relevance to the Bayesian Brain hypothesis.

When learning a Bayesian network from data one might reconstruct the structure of the network, the probability distributions, and even the distributions over hidden variables. Crucially, though, one needs to settle beforehand on the variables and their state space. This is to be contrasted with how generative models in the Bayesian brain hypothesis are actually constructed: Here, one somehow needs to 'learn' new variables and the values they can take, both for potential causes and their observable manifestations. The question then arises *when* a Bayesian learner realizes that the current model is insufficient and new hypotheses should be formed, as well as *what* these hypotheses should look like (Carroll and Kemp, 2013). This process can be coined as *model revision* (Figure

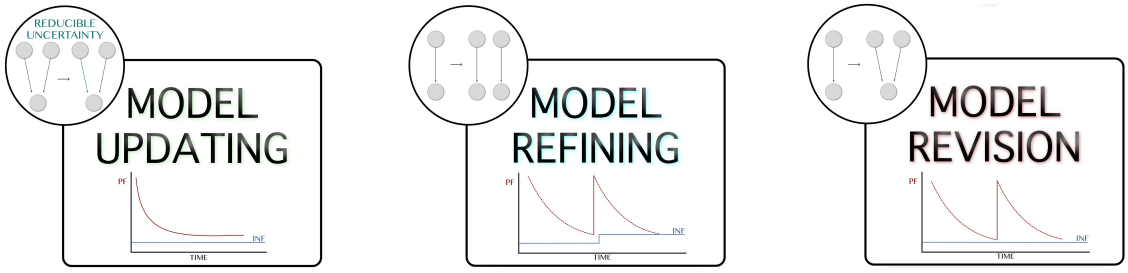


Figure 3: Model updating, model refinement, and model revision processing relative to the prediction error and the amount of irreducible uncertainty. See the main text for explanation of these strategies.

Another vital open problem in the predictive processing account relates to the trade-off between making predictions that are very *detailed* and predictions that are likely to be *correct*. For example, when predicting the outcome of a throw at a bowling lane, a prediction over a distribution containing values like 'pin four will be hit by the ball from the left side and will topple over pins seven and eight' is very detailed, but probably always gives a huge prediction error. On the other hand, a prediction like 'the ball will hit the pins and some will fall' is likely to be correct, but as a prediction not very informative. There are reasons to believe that particular neurotransmitters control this *level of detail* (Pink-Hashkes et al., 2017), but from a more meta-perspective it is completely open how causal Bayesian models can be 'flexible' in their granularity and how algorithms on such models may trade-off information gain and prediction error.

6. Potential Pitfalls

In the previous sections we highlighted several research areas and tentative research questions where the PGM community can substantially contribute to the 'Bayesian Brain' with a potential for considerable impact. Notwithstanding this potential, there are also pitfalls to avoid that are inherent risks of interdisciplinary work, in particular when the research fields have different cultures and tradition and use specific terminology that may be misunderstood. Here we enumerate a few potential pitfalls.

- '**Terminology**' — An informal quiz at the interdisciplinary Lorentz Center workshop 'Perspectives on Human Probabilistic Inference'[1] on the association that participants had with the word 'Bayesian' was illuminative to us. For some participants *Bayesian* was a synonym of *probabilistic*, for others it concerned the semantics of probability distributions (*subjective*, as contrasted with *frequentist*), yet others associated *Bayesian* with *Bayes' rule* for updating distributions. In cognitive science communities, *Bayesian* is often synonymous with *optimal* models and contrasted with *heuristic* explanations. Despite the traditional interpretation of 'Bayesian' as 'subjective degrees of belief' (Jaynes, 2003), it is not uncommon for proponents of the Bayesian Brain hypothesis to have a strong frequentist view on probabilities as describing the objective state of the world (Fiorillo, 2012). Similarly diverse (and sometimes counterintuitive) associations could be elicited for terms like 'prior,' 'uncertainty,' 'information,' and 'structure.' The bottom line is to be aware of potential misunderstandings and to be explicit of one's intended meaning of such terms in communication with neuroscientists.

- '**Culture and tradition**' — In computer science and artificial intelligence, acceptance of a paper to a prestigious conference such as AAAI, UAI, NIPS or STOC is distinctive. Many scholars focus their publication strategy on such conferences, rather than journal papers. In neuroscience, a conference publication is close to irrelevant when it comes to evaluating research output; much more emphasis is put on the impact factor of the journals one is publishing in. Culture and tradition put emphasis on different 'golden standards' of excellence in research, validity of research methodology, and importance of research topics. Awareness of such issues and an open mind may help avoid or solve misunderstandings.

- '**Interdisciplinary**' — Members of interdisciplinary teams have different backgrounds and distinct areas of expertise; that is exactly the main benefit of having interdisciplinary collaborations at all. There is a fine line between 'nitpicking on details' versus 'allowing crucial misconceptions to exist' in interdisciplinary collaborations, and it requires some expertise to see what is important and what not. For example, it is rarely important to insist on the distinction between NP-hardness and NP-completeness of a problem, but the difference between an observation and an intervention in (causal) Bayesian networks may well be important to clarify. Don't assume your neuroscience collaborators share your background, and don't be afraid to ask for clarification about what seems obvious to them. But do understand that a major intellectual effort will be spent on thoroughly understanding each other where this is important for scientific progress.

- '**Selling your work**' — An elegant intractability proof or a new formalization of a verbal theory is typically not sufficient for publication in neuroscience outlets. In order to get published

1. http://www.lorentzcenter.nl/lc/web/2014/627/info.php3?wsid=627&venue=Oort

one should aim to understand the problems that neuroscientists care about, make clear why your contribution is instrumental in solving these problems, and write in a way that connects to their background and expectations. It might be difficult to convince one's departmental chair or (grant) reviewers of the relevance of this work. Our approach is to seek for niches that both allow for a significant PGM contribution *and* solve crucial problems with respect to the Bayesian Brain.

7. Conclusion

Despite the potential pitfalls we identified in the previous section, we strongly believe computer scientists and AI practitioners working in the PGM area can make a vital interdisciplinary contribution to contemporary theoretical neuroscience. With this paper we hope to have given an overview of crucial open problems in the Bayesian Brain hypothesis and a sketch of the contributions that the PGM community can offer. We conclude this paper with this quote from Karl Friston that (probably inadvertently) illustrates the importance of research on probabilistic graphical models for theoretical neuroscience: *Life (. . .) is an inevitable and emergent property of any (ergodic) random dynamical system that possesses a Markov blanket* (Friston, 2013). We would like to invite the community to bring their toolbox of computational and formal modeling and help to advance this fascinating research area — who knows what else may emerge!

Acknowledgments

First ideas for this paper emerged from several discussions at the workshop "Perspectives on Human Probabilistic Inference" organized in May 2014 at the Lorentz Center in Leiden, The Netherlands. A preliminary version of this paper was presented at the 2016 Benelux Conference in Artificial Intelligence.

References

C. Bielza and P. Larrañaga. Bayesian networks in neuroscience: A survey. *Frontiers in Computational Neuroscience*, 8:Article 131, 2014.

L. Buesing, J. Bill, B. Nessler, and W. Maass. Neural dynamics as sampling: A model for stochastic computation in recurrent networks of spiking neurons. *PLoS Computational Biology*, 7(11): e1002211, 2011.

C. D. Carroll and C. Kemp. Hypothesis space checking in intuitive reasoning. In *Proceedings of the 35th Annual Conference of the Cognitive Science Society*, 2013.

A. Clark. Whatever next? Predictive brains, situated agents, and the future of cognitive science. *Behavioral and Brain Sciences*, 36(3):181–204, 2013.

A. Clark. *Surfing Uncertainty: Prediction, Action, and the Embodied Mind*. Oxford University Press, 2015.

V. M. H. Coupé, F. V. Jensen, U. B. Kjærulff, and L. C. van der Gaag. A computational architecture for n-way sensitivity analysis of Bayesian networks. Technical report, Aalborg University, 2000.

P. Dagum and M. Luby. Approximating probabilistic inference in Bayesian belief networks is NP-hard. *Artificial Intelligence*, 60(1):141–153, 1993.

P. Dagum and M. Luby. An optimal approximation algorithm for Bayesian inference. *Artificial Intelligence*, 93:1–27, 1997.

N. Donselaar. Parameterized hardness of active inference. In *Proceedings of PGM'18*, 2018.

C. Fiorillo. Beyond Bayes: On the need for a unified and Jaynesian definition of probability and information within neuroscience. *Information 2012, 3(2)*, 3(2):175–203, 2012.

K. Friston. The free-energy principle: A rough guide to the brain? *Trends in Cognitive Sciences*, 13(7):293–301, 2009.

K. Friston. The free-energy principle: A unified brain theory? *Nature Reviews Neuroscience*, 11 (2):127–138, 2010.

K. Friston. Life as we know it. *Journal of the Royal Society Interface*, 10(86):20130475, 2013.

S. Habenschuss, Z. Jonke, and W. Maass. Stochastic computations in cortical microcircuit models. *PLoS Computational Biology*, 9(11):e1003037, 2013.

T. Hamilton, S. Afshar, A. van Schaik, and J. Tapson. Stochastic electronics: A neuro-inspired design paradigm for integrated circuits. *Proceedings of the IEEE*, 5:843–859, 2014.

Y. Haxhimusa, I. van Rooij, S. Varma, and H. T. Wareham. Resource-bounded problem solving (dagstuhl seminar 14341). *Dagstuhl Reports*, 4(8), 2014.

J. Hohwy. *The Predictive Mind*. Oxford University Press, 2013.

I. Illan, J. Górriz, J. Ramírez, and A. Meyer-Base. Spatial component analysis of MRI data for Alzheimer's disease diagnosis: a Bayesian network approach. *Frontiers in Computational Neuroscience*, 8:156, 2014.

E. Jaynes. How does the brain do plausible reasoning? In G. J. Erickson and C. R. Smith, editors, *Maximum-Entropy and Bayesian Methods in Science and Engineering*, 1988.

E. Jaynes. *Probability Theory: The Logic of Science*. Cambridge University Press, 2003.

R. Jeffrey. *The Logic of Decision*. University of Chicago Press, 1965.

J. W. Kay and W. A. Phillips. Coherent infomax as a computational goal for neural systems. *Bulletin of Mathematical Biology*, 73(2):344–372, 2011.

J. Kiverstein, M. Miller, and E. Rietveld. The feeling of grip: Novelty, error dynamics, and the predictive brain. *Synthese*, 2017. doi: https://doi.org/10.1007/s11229-017-1583-9.

D. Knill and A. Pouget. The Bayesian brain: the role of uncertainty in neural coding and computation. *Trends in Neurosciences*, 27(12):712–719, 2004.

J. Kwisthout. Minimizing relative entropy in hierarchical predictive coding. In L. van der Gaag and A. Feelders, editors, *Proceedings of PGM'14*, LNCS 8754, pages 254–270, 2014.

J. Kwisthout. Tree-width and the computational complexity of MAP approximations in Bayesian networks. *Journal of Artificial Intelligence Research*, 53:699–720, 2015.

J. Kwisthout. Approximate inference in Bayesian networks: Parameterized complexity results. *International Journal of Approximate Reasoning*, 93:119–131, 2018.

J. Kwisthout and I. van Rooij. Predictive processing and the Bayesian brain: Intractability hurdles that are yet to overcome. *Computational Brain and Behavior*, under review.

J. Kwisthout, H. Bekkering, and I. van Rooij. To be precise, the details don't matter: On predictive processing, precision, and level of detail of predictions. *Brain and Cognition*, 112:84–91, 2017.

M. L. Littman, J. Goldsmith, and M. Mundhenk. The computational complexity of probabilistic planning. *Journal of Artificial Intelligence Research*, 9:1–36, 1998.

W. Maass. Neural computation: a research topic for theoretical computer science? Some thoughts and pointers. In *Bulletin of the European Association for Theoretical Computer Science (EATCS)*, volume 72. 2000.

W. Maass. Noise as a resource for computation and learning in networks of spiking neurons. *Proceedings of the IEEE*, 102(5):860–880, 2014.

D. Marr. *Vision: A computational investigation into the human representation and processing of visual information*. New York: Freeman, 1982.

B. Mihaljević, C. Bielza, R. Benavides-Piccione, J. DeFelipe, and P. Larrañaga. Multi-dimensional classification of GABAergic interneurons with Bayesian network-modeled label uncertainty. *Frontiers in Computational Neuroscience*, 8:150, 2014.

M. Otworowska, J. Kwisthout, and I. van Rooij. Counter-factual mathematics of counterfactual predictive models. *Frontiers in Consciousness Research*, 5:801, 2014.

M. Otworowska, J. Riemens, C. Kamphuis, P. Wolfert, L. Vuurpijl, and J. Kwisthout. The robohavioral methodology: Developing neuroscience theories with FOES. In *Proceedings of the 27th Benelux Conference on AI (BNAIC'15)*, 2015.

J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge: MIT Press, 2000.

D. Pecevski, L. Bueling, and W. Maass. Probabilistic inference in general graphical models through sampling in stochastic networks of spiking neurons. *PLoS Computational Biology*, 7(12):1–25, 2011.

W. Phillips. Cognitive functions of intracellular mechanisms for contextual amplification. *Brain and Cognition*, 12:39–53, 2017.

S. Pink-Hashkes, I. van Rooij, and J. Kwisthout. Perception is in the details: A predictive coding account of the psychedelic phenomenon. In *Proceedings of the 39th Annual Meeting of the Cognitive Science Society*, pages 2907–2912, 2017.

S. Schoenmakers, U. Güçlü, M. van Gerven, and T. Heskes. Gaussian mixture models and semantic gating improve reconstructions from human brain activity. *Frontiers in Computational Neuroscience*, 8:173, 2015.

J. B. Tenenbaum, C. Kemp, T. Griffiths, and N. Goodman. How to grow a mind: Statistics, structure, and abstraction. *Science*, 331:1279–1285, 2011.

C. Thornton. Predictive processing simplified: The infotropic machine. *Brain and Cognition*, 112: 13–24, 2017.

L. C. van der Gaag and H. L. Bodlaender. On stopping evidence gathering for diagnostic Bayesian networks. In *Proceedings of the Eleventh European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, volume 6717 of *LNCS*, pages 170–181, 2011.

H. von Helmholtz. *Handbuch der Physiologischen Optik*. Leipzig: Leopold Voss, 1867.

A. J. Yu and P. Dayan. Uncertainty, neuromodulation, and attention. *Neuron*, 46:681–692, 2005.

Branch and Bound for Continuous Bayesian Network Structure Learning

Joe Suzuki J-SUZUKI@SIGMATH.ES.OSAKA-U.AC.JP
Osaka University, Japan.

Abstract

We consider the Bayesian network structure learning (BNSL) problem when the variables are continuous. To this end, we construct a dynamic programming (DP) -based algorithm, and consider applying a branch and bound (B&B) approach to speed up computations. Although B&B has been applied to discrete BNSL in the literature, neither DP nor B&B have not been considered for continuous BNSL. Our scores are information criteria in the form $-log(likelihood) + K * d(N)$, where K and $d(N)$ are the size of parent set and a function of sample size N (if $d(N) = \frac{1}{2}\log N$, then the information criterion is BIC). We derive a lower bound for the B&B framework, and did experiments for various $d(N)$, N, and p (the number of variables). The surprising news is that the proposed B&B is considerably efficient: $5 \sim 10$ times faster and $20 \sim 100$ times faster for $p = 20$ and for $p = 25$, respectively, compared with when no B&B is applied.

Keywords: branch and bound, Bayesian networks, structure learning, BIC.

1. Introduction

We consider learning stochastic relations among variables from data. If we mean by the relations conditional independence (CI) among variables, and if we express them via a directed acyclic graph (DAG), then such a graphical model will be a Bayesian network (BN) (Pearl, 1988). In general, a BN is defined by its structure and parameters, i.e., its topology of nodes and edges and the conditional probabilities of variables given other variables.

There are several approaches for Bayesian network structure learning (BNSL). We may test each CI statement between two variable sets given another variable set with the three disjoint sets using a heuristic such as the PC algorithm (Spirtes et al., 1993). In this paper, however, we focus on score-based approaches such as maximizing the posterior probability of a selected structure based on the prior probability and data (Cooper and Herskovits, 1992), or minimizing the description length (MDL) (Rissanen, 1978) of data for a selected structure (Suzuki, 1993): given data, we compute its score for each structure and select a structure with the optimal value.

In this paper, we focus on the BNSL problem for continuous variables.

BNSL consists of finding the optimal parent sets and ordering of the variables. We note that as the number of variables grows, the computation exponentially increases (Chickering et al., 2003). For many years, several authors of BNSL have been considering pruning the search space when searching the optimal parent sets in a depth-first manner. Suzuki (1996) proposed a pruning rule for the MDL principle to reduce the computation; Tian (2000) proposed variants of Suzuki (1996); Campos and Ji (2011) pointed out that finding the optimal parent sets w.r.t. the MDL principle takes at most polynomial time of p when the sample size N is a constant; Campos and Ji (2011) also proposed a pruning rule for the BDeu; and Suzuki (2016) proposed a pruning rule for maximizing the posterior probability based on Jeffreys' prior. Recently, Suzuki (2017) proved that BDeu based BNSL is not regular; and Suzuki and Kawahara (2017) claimed that it is the main reason of why

the B&B for BDeu proposed by Campos and Ji (2011) is not efficient. and proposed a framework containing the B&B approaches of regular BNSL.

However, no B&B for BNSL with continuous variables have been considered thus far. Even the unified framework proposed by (Silander and Myllymaki, 2006; Singh and Moore, 2005; Ott et al., 2004) was not used for continuous BNSL. The commonly used approach to continuous BNSL is learning an undirected graph structure first and estimating the directions (v-structure identification) later in which the PC algorithm (Spirtes et al., 1993) is often used for the first part, However, the two-step approach works only when the sample size is large because the second part assumes that the first part which depends on some CI test is correct.

In this paper, we apply the unified framework that has been used only for discrete BNSL to the continuous counterpart. The score will be a wide range of information criteria that contain AIC (Akaike, 1973) and BIC (Schwarz, 1978), etc.

The main issue in this paper is whether the B&B approach works for continuous BNSL. In this paper, we derive a lower bound for B&B, and construct an algorithm that contains the bound in the dynamic programming framework.

From our experiments, we find that the proposed B&B runs $5 \sim 10$ times faster and $20 \sim 100$ times faster for $p = 20$ and for $p = 25$, respectively, compared with when no B&B is applied. Also, we will see that the efficiency does not decay so much when $d(N)$ becomes small as the discrete B&B ones do. Similar phenomena was examined also by real data such as Hitters and breastcancer data sets.

Some might say that the proposed procedure is only for Gaussian BNs rather than (general) continuous BNs. In fact, the score is derived assuming the underlying distribution is Gaussian. However, the score in the form of an information criterion is popular and is used in many cases. This paper proposes a computation procedure and can be used even when no Gaussian distribution is assumed.

This paper is organized as follows: Section 2 introduces background material for understanding the results in this paper: linear regression and information criteria, a formulation of the Bayesian network structure learning (BNSL) problem that consists of two parts, a dynamic programming framework of finding parent sets, and a B&B approach for BNSL with discrete variables. Section 3 proposes a novel B&B algorithm for BNSL with continuous variables: derive a lower bound of the B&B for saving the computation and construct a proposed algorithm based on the bound. Section 4 shows the results of experiments for various information criteria, and applications to real-world datasets. Section 5 concludes the discussion and raises future research directions.

2. Preliminaries

In this section, we introduce background material for understanding the results in this paper.

Suppose we have N samples $\{(x_{i,1}, \cdots, x_{i,p})\}_{i=1}^N$ from p (continuous) variables, where each $x_{i,j} \in \mathbb{R}$ is a realization of variable X_j, $j = 1, \cdots, p$, and they are related by the equations: for $k = 1, \cdots, p$

$$X_k = \sum_{j=1}^{k-1} \beta_{k,j} X_j + \epsilon_k \ ,$$

where $\{\beta_{k,j}\}_{j=1}^{k-1}$ are unknown constants, and variable ϵ_k is independent of $\epsilon_1, \cdots, \epsilon_{k-1}$. We refer $\pi_k := \{j | \beta_{k,j} \neq 0\}$ as to the parent set of variable X_k.

We do not know the order among the p variables implied by the equation above and parent sets $\pi_1, \cdots, \pi_p$ a priori but need to estimate them from the data. We refer to this problem as BNSL with continuous variables.

2.1 Linear Regression and Information Criteria

Suppose we have N samples $\{(x_{i,1}, \cdots, x_{i,p}, y_i)\}_{i=1}^N$ from p variables, where $(x_{i,1}, \cdots, x_{i,p}) \in \mathbb{R}^p$ and $y_i \in \mathbb{R}$ are realizations of variables $(X_1, \cdots, X_p)$ and Y, respectively, and they are related by the equation:

$$Y = \sum_{j=1}^{p} \beta_j X_j + \epsilon$$

with $\{\beta_j\}_{j=1}^p$ and ϵ being constants and a random variable, respectively.

To find the parent set $\pi = \{j | \beta_j \neq 0\}$ from the data, we often compare the values of an information criterion of the form

$$IC := N \log \hat{\sigma}^2 + |\pi| d(N)$$

for the candidate parent sets to choose the one that minimizes the score, where $\hat{\sigma}^2$ is the residual sum of squares for parent set π, $|S|$ is the cardinality of set S, and $d(N)$ is a function of N. More precisely, we have

$$\hat{\sigma}^2 := \frac{1}{N-1} \sum_{i=1}^{N} (y_i - \sum_{j \in \pi} \hat{\beta}_j x_{i,j})^2 \,, \tag{1}$$

where $\hat{\beta}_j$ are estimates of the β_j, $j = 1, \cdots, p$.

The information criteria we consider are Akaike's information criterion (AIC), the Hannan and Quinn (HQ), and the Baysian information criterion (BIC) for $d(N) = 1$, $d(N) = \log \log N$, and $d(N) = \frac{1}{2} \log N$, respectively (Akaike, 1973; Hannan and Quinn, 1979; Schwarz, 1978). However, any $d(N)$ can be used as long as it takes nonnegative values and increases monotonically in N. It is known that the parent set π is estimated correctly in probability and almost surely if

$$\lim_{N \to \infty} d(N) = \infty \text{ and } \lim_{N \to \infty} \frac{d(N)}{N} = 0$$

and if

$$\lim_{N \to \infty} \frac{d(N)}{\log \log N} = \infty \text{ and } \lim_{N \to \infty} \frac{d(N)}{N} = 0 \,, \tag{2}$$

respectively (Suzuki, 2006). Note that BIC satisfies both conditions while AIC satisfies neither, and that HQ has the minimum $d(N)$ that satisfies (2).

2.2 A Unified Approach for BNSL

We reduce BNSL to finding the parent sets $\pi_1, \cdots, \pi_p$ that minimize the sum of the scores

$$IC(k \sim \pi_k) := N \log \hat{\sigma}_k^2 + |\pi_k| d(N) \tag{3}$$

over $k = 1, \cdots, p$, where the parameters in (1) are replaced by

$$\hat{\sigma}_k^2 := \frac{1}{N-1} \sum_{i=1}^{N} (x_{i,k} - \sum_{j \in \pi_k} \hat{\beta}_{k,j} x_{i,j})^2 \,.$$

Then, we consider minimizing the quantity $IC(k \sim \pi_k)$ w.r.t. $\pi_k \subseteq \{1, 2, \cdots, k-1\}$ for each $k = 1, \cdots, p$.

In this problem, however, we do not know the order among the variables a priori and estimate it by minimizing the total score.

Let $(a_1, \cdots, a_p)$ be a permutation of $(1, \cdots, p)$. We extend k and $\{1, 2, \cdots, k-1\}$ into a_k and $\{a_1, \cdots, a_{k-1}\}$ in the definition (3), respectively, and define the value of $IC(a_k \sim \pi_k)$. We find the permutation $(a_1, \cdots, a_p)$ that minimizes

$$\sum_{k=1}^{p} \min_{\pi_k \subseteq \{a_1, \cdots, a_{k-1}\}} IC(a_k \sim \pi_k) \, . \tag{4}$$

For example, if $p = 3$, then we compute the following twelve

$$\begin{array}{cccc} \min_{\pi \subseteq \{\}} IC(1 \sim \pi) & \min_{\pi \subseteq \{\}} IC(2 \sim \pi) & \min_{\pi \subseteq \{\}} IC(3 \sim \pi) & \min_{\pi \subseteq \{1\}} IC(2 \sim \pi) \\ \min_{\pi \subseteq \{1\}} IC(3 \sim \pi) & \min_{\pi \subseteq \{2\}} IC(3 \sim \pi) & \min_{\pi \subseteq \{2\}} IC(1 \sim \pi) & \min_{\pi \subseteq \{2,3\}} IC(1 \sim \pi) \\ \min_{\pi \subseteq \{3,1\}} IC(2 \sim \pi) & \min_{\pi \subseteq \{1,2\}} IC(3 \sim \pi) & \min_{\pi \subseteq \{1\}} IC(3 \sim \pi) & \min_{\pi \subseteq \{2\}} IC(3 \sim \pi) \end{array} \tag{5}$$

and compare the following six

$$\begin{array}{l} \min_{\pi_1 \subseteq \{\}} IC(1 \sim \pi_1) + \min_{\pi_2 \subseteq \{1\}} IC(2 \sim \pi_2) + \min_{\pi_3 \subseteq \{1,2\}} IC(3 \sim \pi_3) \\ \min_{\pi_1 \subseteq \{\}} IC(1 \sim \pi_1) + \min_{\pi_2 \subseteq \{1\}} IC(3 \sim \pi_2) + \min_{\pi_3 \subseteq \{1,3\}} IC(2 \sim \pi_3) \\ \min_{\pi_1 \subseteq \{\}} IC(2 \sim \pi_1) + \min_{\pi_2 \subseteq \{2\}} IC(1 \sim \pi_2) + \min_{\pi_3 \subseteq \{1,2\}} IC(3 \sim \pi_3) \\ \min_{\pi_1 \subseteq \{\}} IC(2 \sim \pi_1) + \min_{\pi_2 \subseteq \{2\}} IC(3 \sim \pi_2) + \min_{\pi_3 \subseteq \{2,3\}} IC(1 \sim \pi_3) \\ \min_{\pi_1 \subseteq \{\}} IC(3 \sim \pi_1) + \min_{\pi_2 \subseteq \{3\}} IC(1 \sim \pi_2) + \min_{\pi_3 \subseteq \{1,3\}} IC(2 \sim \pi_3) \\ \min_{\pi_1 \subseteq \{\}} IC(3 \sim \pi_1) + \min_{\pi_2 \subseteq \{3\}} IC(2 \sim \pi_2) + \min_{\pi_3 \subseteq \{2,3\}} IC(1 \sim \pi_3) \end{array} \tag{6}$$

for $(a_1, a_2, a_3) = (1, 2, 3), (1, 3, 2), (2, 1, 3), (2, 3, 1), (3, 1, 2)$, and $(3, 2, 1)$. Thus, we will find optimal parent sets. For example, if the fourth in (6) is the smallest among the six, then the π_1, π_2, π_3 that minimize $IC(2 \sim \pi_1)$, $IC(3 \sim \pi_2)$, $IC(1 \sim \pi_3)$ will be the parent sets of X_2, X_3, X_1, respectively.

In general, BNSL is classified into two subproblems (Silander and Myllymaki, 2006; Singh and Moore, 2005; Ott et al., 2004), which can be investigated separately:

1. compute $\min_{\pi \subseteq S} IC(a \sim \pi)$ for each pair of $a \in \{1, \cdots, p\}$ and $S \subseteq \{1, \cdots, p\} \backslash \{a\}$, where $\pi \subseteq S$ is the parent set; and

2. using the $p2^{p-1}$ values computed in the first step, obtain the permutation $(a_1, \cdots, a_p)$ of $(1, \cdots, p)$ that minimizes (4).

The two-step framework directly identifies an optimal DAG, and does not require learning an undirected graph structure first and estimating the directions later. For example, if $p = 3$, then we compute $p2^{p-1} = 12$ quantities as in (5) and compare $p! = 6$ quantities as in (6).

On the other hand, we can consider the second problem as the shortest path problem. In fact, $\min_{\pi_k \subseteq \{a_1, \cdots, a_{k-1}\}} IC(a_k \sim \pi_k)$ is the shortest path to a_k when the nodes $a_1, \cdots, a_k$ and the distances $IC(a \sim S)$ obtained in the first problem are given (Yuan and Malone, 2013). Thus, whether the BNSL is either discrete or continuous does not matter to the second problem while it does to the first.

In this paper, we mainly consider the first problem.

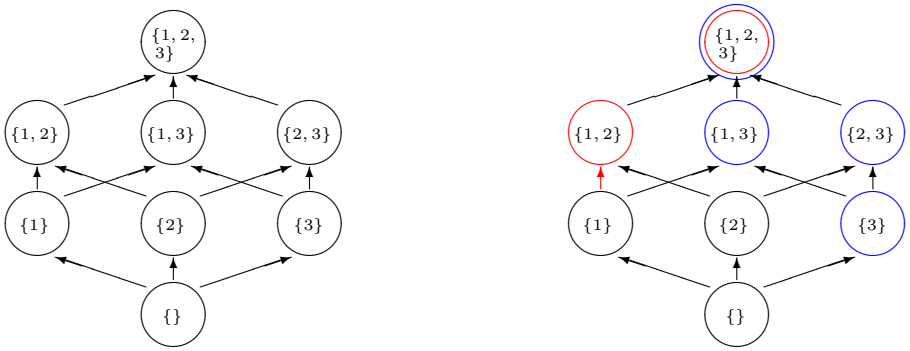


Figure 1: Left: The ordered graph from $\{\}$ to $\{1,2,3\}$: compute $IC^*(S)$ for $S \subseteq \{1,2,3\}$ and find their aseociated parent sets in a bottom-up manner. Right: If $IC^*(\{1\})$ is a lower-bound of $IC(\{1,2\})$, then we do not have to compute $IC(\{1,2\})$ and $IC(\{1,2,3\})$ in red circles; and if $IC^*(\{\})$ is a lower-bound of $IC(\{3\})$, then we do not have to compute $IC(\{3\}), IC(\{1,3\}), IC(\{2,3\})$ and $IC(\{1,2,3\})$ in blue circles.

2.3 Finding the Parent Sets via Dynamic Programming

In this subsection, we fix $a_k = a$ and drop suffix k to consider the minimization of $IC(a \sim \pi)$ w.r.t. $\pi \subseteq S$ for all $S \subseteq \{1, \cdots, p\} \backslash \{a\}$.

Let $IC(S) := IC(a \sim S)$ and $IC^*(S) := \min_{\pi \subseteq S} IC(a \sim \pi)$. Then, we have (Silander and Myllymaki, 2006)

$$IC^*(S) = \min\{IC(S), \min_{b \in S} IC^*(S \backslash \{b\})\} \ . \tag{7}$$

We see that the values of $IC^*(S)$ for $S \subseteq \{1, \cdots, p\} \backslash \{a\}$ can be obtained via an ordered graph (Figure 1) in a bottom up manner. Suppose that $p = 4$ and $a = 4$, thus $S \subseteq \{1,2,3\}$ as in Figure 1 (Left). At first, we find $\pi_1 = \{\}$ and compute $IC^*(\{\}) = IC(\{\})$. Then, from (7), we have

$$IC^*(\{1\}) = \min\{IC(\{1\}), IC^*(\{\})\} = \min\{IC(\{1\}), IC(\{\})\} \ .$$

Similarly, we obtain $IC^*(\{2\})$ and $IC^*(\{3\})$. Furthermore, from (7), we obtain

$$IC^*(\{1,2\}) = \min\{IC(\{1,2\}), \min\{IC^*(\{1\}), IC^*(\{2\})\}\}$$

and other two valus ($IC^*(\{2,3\})$, $IC^*(\{3,1\})$). Finally, we obtain

$$IC^*(\{1,2,3\}) = \min\{IC(\{1,2,3\}), \min\{IC^*(\{1,2\}), IC^*(\{2,3\}), IC^*(\{3,1\})\}\} \ .$$

Note that $IC^*(S) := \min_{S' \subseteq S} IC(S')$ holds for all $S \subseteq \{1,2,3\}$.

2.4 Branch and Bound for BNSL with Discrete Variables

In order to estimate (4), we need to compute the $p2^{p-1}$ values for the subsets of parents. In this subsection, we consider reducing the computation using the so-called B&B technique. B&B approaches, even though they are more cimputationally efficient, are guaranteed to find a globally optimal solution.

Suppose that there exist a subset $S \subseteq \{1, \cdots, p\} \backslash \{a\}$ and its element $b \in S$ such that in (7),

$$IC^*(S \backslash \{b\}) \leq IC(S') \tag{8}$$

for $S' \supseteq S$. Then, we do not have to compute the value $IC(S')$ for $S' \supseteq S$, and can conclude $IC^*(S') \leq IC^*(S \backslash \{b\})$ for $S' \supseteq S$.

For example, in Figure 1 (Right), if the valus of $IC(S)$ is bounded below by c for $S \supseteq \{1, 2\}$ and we find $IC^*(\{1\}) \leq c$, then we can conclude $IC^*(\{1\}) \leq IC(S)$ and $IC^*(S) \leq IC^*(\{1\})$ for $S = \{1, 2\}, \{1, 2, 3\}$. Note that we cannot tell from this whether $IC^*(S) = IC^*(\{1\})$ for $S = \{1, 2\}, \{1, 2, 3\}$. For example, if $c' := IC^*(\{2\}) < c$, we obtain $IC^*(\{1, 2\}) = c'$; and if $c'' := \min\{IC^*(\{1, 3\}, IC^*(\{2, 3\})\}$ is less than c and c', we obtain $IC^*(\{1, 2, 3\}) = c''$.

To apply B&B to BNSL, we need to derive a lower bound formula. The first lower bound was proposed by Suzuki (1996) for discrete variables, and several authors considered variants afterward (Tian, 2000; Campos and Ji, 2011). For the BDeu scores (Buntine, 1991; Ueno, 2008), (Campos and Ji, 2011) and (Cussens and Bartlett, 2015) derived lower bounds. Recently, (Suzuki, 2017; Suzuki and Kawahara, 2017) proved that BDeu is not a regular BNSL and proposed a novel B&B method for regular BNSL. However, thus far, the B&B technique has been proposed only for BNSL with discrete variables.

Let α_j be the number of possible values that X_j takes for $j = 1, \cdots, p$. Suppose that we find the parent set π of variable X_a with $a \in \{1, \cdots, p\}$ over $\pi \subseteq S \subseteq \{1, \cdots, p\} \backslash \{a\}$. In the discrete settings, the negated log likelihood is proportioinal to the empirical conditional entropy $H(\pi)$ of X_a given $\{X_k | k \in \pi\}$, and the number of parameters will be $K(\pi) := (\alpha_a - 1) \prod_{k \in \pi} \alpha_k$. Thus, the description length (Rissanen, 1978) will be $H(\pi) + \dfrac{K(\pi)}{2} \log n$.

Suppose that the inequality in

$$IC^*(S \backslash \{b\}) := \min_{\pi \subseteq S \backslash \{b\}} 2\{H(\pi) + \frac{K(\pi)}{2} \log n\} \leq K(S) \log n$$

holds for some $b \in S$. Then, since $H(S') \geq 0$ and $K(S') \geq K(S)$ for $S' \supseteq S$, we have (8):

$$IC(S') := 2H(S') + K(S') \log n \geq K(S) \log n \geq IC^*(S \backslash \{b\})$$

for any $S' \supseteq S$ (Suzuki, 1996).

For example, in Figure 1 (Right), if $IC^*(\{1\}) \leq K(\{1, 2\}) \log n$, then we can conclude $IC(S) \geq IC^*(\{1\})$ and $IC^*(S) \leq IC^*(\{1\})$ for $S = \{1, 2\}, \{1, 2, 3\}$ (see the red circle).

3. Branch and Bound for BNSL with Continuous Variables

In this section, we propose a novel B&B algorithm for BNSL with continuous variables.

3.1 Proposed Pruning Rule

In this subsection, we derive a lower bound for applying B&B to BNSL with continuous variables. Without loss of generality, we assume that $a = p$ and $S \subseteq \{1, \cdots, p-1\}$, and define

$$IC(p \sim \pi) := N \log \hat{\sigma}_\pi^2 + |\pi| d(N) \ ,$$

where $\hat{\sigma}_\pi^2 := \dfrac{1}{N-1}\sum_{i=1}^{N}(x_{i,p} - \sum_{j\in\pi}\hat{\beta}_{p,j}x_{i,j})^2$ for $\pi \subseteq S$.

Let $IC(S) := IC(p \sim S)$ and $IC^*(S) := \min_{\pi\subseteq S} IC(p\sim\pi)$. If $IC^*(S\backslash\{b\}) \leq \min_{S'\supseteq S} IC(S')$ for $S \subseteq \{1,\cdots,p-1\}$ and $b \in S$, we do not have to compute any $IC(S')$ for $S' \supseteq S$, and can conclude that $IC^*(S') \leq IC^*(S\backslash\{b\})$ for $S' \supseteq S$. We give a lower-bound of $\min_{S'\supseteq S} IC(S')$ as follows:

Proposition 1 $IC(S') \geq N\log\sigma^2_{\{1,\cdots,p-1\}} + |S|d(N)$ for $S' \supseteq S$.

Proof. Noting $S \subseteq S' \subseteq \{1,\cdots,p-1\}$, we have $\hat{\sigma}^2_{S'} \geq \hat{\sigma}^2_{\{1,\cdots,p-1\}}$ and $|S'| \geq |S|$, which implies the claim:

$$IC(S') = N\log\hat{\sigma}^2_{S'} + |S'|d(N) \geq N\log\sigma^2_{\{1,\cdots,p-1\}} + |S|d(N)\ .$$

For example, if $p = 4$, then the maximum log-likelihood is

$$L := \max_{\beta_{4,1},\beta_{4,2},\beta_{4,3}} N\log\{\frac{1}{N-1}\sum_{i=1}^{N}(x_{i,4} - \sum_{j=1}^{3}\beta_{4,j}x_{i,j})^2\}\ ,$$

and for each subset S of $\{1,2,3\}$, the lower-bound in Proposition 1 is $L + |S|d(N)$.

3.2 Proposed Algorithm

In this subsection, based on Proposition 1, we construct an algorithm that finds the parent sets $\pi(S)$ for all $S \subseteq \{1,\cdots,p-1\}$ that minimize $IC(p\sim\pi)$ w.r.t. $\pi \subseteq S$, with less computation. We show the procedure in Algorithm 1. The definitions of $\hat{\sigma}^2_S$, $IC(S)$, and $IC*(S)$ for $S \subseteq \{1,\cdots,p-1\}$ are given in the previous subsection.

Algorithm 1
Input $\{(x_{i,1},\cdots,x_{i,p})\}_{i=1}^N$, **Output** $\pi(S)$, $S \subseteq \{1,\cdots,p-1\}$

In the ascending order[1] of $S \subseteq \{1,\cdots,p-1\}$, we set $cut(S)$:=FALSE for $S \subseteq \{1,\cdots,p-1\}$ and take the following steps for each S.

1. if $cut(S\backslash\{b\}) =$ TRUE for any $b \in S$, then $cut(S) :=$ TRUE;
2. if $b^* := \arg\min_{b\in S} IC^*(S\backslash\{b\})$, then $\begin{cases} \pi(S) := \pi(S\backslash\{b^*\}) \\ IC^*(S) := IC^*(S\backslash\{b^*\}) \end{cases}$
3. if $cut(S) =$ FALSE, then
   (a) if $IC^*(S\backslash\{b^*\}) < N\log\hat{\sigma}^2_{\{1,\cdots,p-1\}} + |S|d(N)$, then $cut(S) :=$ TRUE;
   (b) else if $IC(S) < IC^*(S)$, then $\begin{cases} \pi(S) := S \\ IC^*(S) := IC(S) \end{cases}$

If the condition in the pruning rule in 3(a) is met for S and b^*, then $cut(S) :=$ TRUE will be set. Then, $cut(S') :=$ TRUE will also be set for all $S' \supseteq S$ as in Step 1. Step 2 computes

1. If $S' \subsetneq S$, then S' should be executed before S.

$\min_{b \in S} IC(S \backslash \{b\})$ and its associated parent set. The values of $\pi(S)$ and $IC^*(S)$ in Step 2 are updated in Step 3(b) only if $IC(S) < IC^*(S)$. Once $cut(S)$:=TRUE is set for S, step 3 will not be executed ($IC(S')$ will not be computed) for any $S' \supseteq S$. However, it is possible that the value of $IC^*(S')$ may become lower in Step 2.

Theorem 1 Algorithm 1 finds the parent sets $\pi(S)$ for all $S \subseteq \{1, \cdots, p-1\}$ that minimize $IC(p \sim \pi)$ w.r.t. $\pi \subseteq S$.

Note that the value $N \log \hat{\sigma}^2_{\{1,\cdots,p-1\}}$ can be obtained outside the loop at the beginning of the algorithm, and that checking the pruning rule requires only trivial computation. So, we expect that the additional overhead for B&B is negligible.

If we remove Steps 1 and 3(a) from Algorithm 1, we obtain a pure dynamic programming procedure to obtain $\pi(S)$, $S \subseteq \{1, \cdots, p-1\}$ without B&B. By embedding those steps, we can avoid computating $IC(S)$ for some $S \subseteq \{1, \cdots, p-1\}$.

4. Experiments

In this section, we show some results on experiments. Because an optimal solution will be obtained even if the B&B is applied to the dynamic programming. we evaluate Algorithm 1 only by its efficiency.

4.1 Using Various Information Criteria

We apply $d(N) = 1$ (AIC Akaike (1973)), $d(N) = \log \log N$ (HQ Hannan and Quinn (1979)), $d(N) = \frac{1}{2} \log n$ (BIC Schwarz (1978), MDL Rissanen (1978)), and $d(N) = \sqrt{N}$ to artificial data $\{(x_1, \cdots, x_p)\}_{i=1}^N$ with $N = 100, 200, 400$ and $p = 11, 16, 21$. For simplicity, we obtained parent sets for $p \sim S$ with $S \subseteq \{1, \cdots, p-1\}$.

We first generate data for experimnts. For $i = 1, \cdots, p$, we obtain $x_i \in \mathbb{R}^N$ as follows:

1. generate $\beta_1, \cdots, \beta_{i-1} \sim N(0, 1)$ and $\epsilon_1, \cdots, \epsilon_N \sim N(0, 1)$.
2. $x_i = \alpha_i \sum_{j=1}^{i-1} \beta_j x_j + \epsilon \in \mathbb{R}^N$ with $x_1, \cdots, x_{i-1} \in \mathbb{R}^N$ and $\epsilon = \begin{bmatrix} \epsilon_1 \\ \vdots \\ \epsilon_N \end{bmatrix} \in \mathbb{R}^N$

where $\alpha_i > 0$, $i = 1, \cdots, p$, are constants. In particular, we consider three cases: $\alpha_1 = \cdots = \alpha_p = 1$ (Table 1 (a)), $\alpha_1 = \cdots = \alpha_{p-1} = 1, \alpha_p = 0.3$ (Table 1 (b)), and $\alpha_1 = \cdots = \alpha_{p-1} = 0.3, \alpha_p = 1$ (Table 1 (c)). Because the optimal solution is always obtained, we evaluate Algorithm 1 only in terms of efficiency. The actual execution time and the number of subsets $S \subseteq \{1, \cdots, p-1\}$ such that $IC(s)$ is actually computed divided by 2^{p-1}. We obtain insights from the experiments for $d(N) = 1, \log \log N, \log N, \sqrt{N}$, $N = 100, 250, 1000$, and $p = 16, 21, 26$. The algorithm is executed via Rcpp (Eddelbuettel, 2013): each compiled Rcpp procedure runs as an R function almost as fast as when the same procedure runs as a C++ function. The CPU we used in the experiments was Core M-5Y10(Broadwell)/800MHz/2.

From Table 1 which contains all the numerical results in this subsection, we find that the execution is considerably efficient for most of the cases. For example, for the case $p = 25$, $N = 100$,

Table 1: Ratio and Time for $N = 100, 200, 400$, $p = 16, 21, 26$, and $d(N) = 1, \log\log N, \frac{1}{2}\log N$, and $\sqrt{N}$. The upper and lower figures are the ratios (how often $IC(S)$ was computed divided by 2^{p-1}) and actual times (seconds). We measured those values for three cases: (a) $\alpha_1 = \cdots = \alpha_p = 1$, (b) $\alpha_1 = \cdots = \alpha_{p-1} = 1$, $\alpha_p = 0.3$, (c) $\alpha_1 = \cdots = \alpha_{p-1} = 0.3$, $\alpha_p = 1$.

(a) $\alpha_1 = \cdots = \alpha_p = 1$

	$d(N)=1$			$d(N)=\log\log N$			$d(N)=\frac{1}{2}\log N$			$d(N)=\sqrt{N}$		
p	100	250	1000	100	250	1000	100	250	1000	100	250	1000
16	0.114	0.129	0.155	0.110	0.125	0.154	0.099	0.114	0.152	0.052	0.068	0.114
	0.32	0.46	1.05	0.29	0.37	0.96	0.25	0.36	1.0	0.25	0.30	0.75
21	0.044	0.052	0.056	0.049	0.051	0.055	0.041	0.050	0.055	0.021	0.032	0.044
	0.44	13.87	20.80	10.20	11.71	21.92	9.91	11.94	21.48	9.41	0.43	7.79
26	0.0026	0.0033	0.0038	0.0024	0.0031	0.0038	0.0022	0.0029	0.0037	0.0009	0.0015	0.0025
	467.24	390.25	396.79	563.92	468.81	564.89	524.23	530.17	452.89	517.27	497.81	547.59

(b) $\alpha_1 = \cdots = \alpha_{p-1} = 1$, $\alpha_p = 0.3$

	$d(N)=1$			$d(N)=\log\log N$			$d(N)=\frac{1}{2}\log N$			$d(N)=\sqrt{N}$		
p	100	250	1000	100	250	1000	100	250	1000	100	250	1000
16	0.233	0.261	0.241	0.228	0.261	0.240	0.224	0.258	0.238	0.171	0.207	0.216
	0.44	0.77	1.95	0.37	0.73	1.78	0.41	0.79	2.31	0.52	0.83	2.18
21	0.021	0.029	0.023	0.021	0.026	0.023	0.019	0.025	0.022	0.0100	0.0151	0.0176
	9.45	11.93	17.16	9.25	14.54	15.72	9.08	11.17	13.86	13.09	15.72	18.31
26	0.0036	0.0044	0.0035	0.0034	0.0043	0.0035	0.0032	0.0041	0.0035	0.00145	0.0240	0.0027
	452.86	443.77	565.45	538.72	494.83	571.98	491.91	519.56	421.18	573.37	514.93	433.80

(c) $\alpha_1 = \cdots = \alpha_{p_1} = 0.3$, $\alpha_p = 1$

	$d(N)=1$			$d(N)=\log\log N$			$d(N)=\frac{1}{2}\log N$			$d(N)=\sqrt{N}$		
p	100	250	1000	100	250	1000	100	250	1000	100	250	1000
16	0.432	0.428	0.410	0.422	0.419	0.408	0.402	0.402	0.403	0.193	0.266	0.335
	1.08	0.97	3.33	0.64	1.00	2.78	0.59	0.95	2.88	0.34	0.63	2.5
21	0.247	0.273	0.265	0.230	0.266	0.263	0.204	0.253	0.259	0.0514	0.105	0.178
	27.14	29.86	92.83	18.9	33.43	75.06	16.55	35.64	74.59	10.34	14.79	54.94
26	0.179	0.215	0.194	0.160	0.200	0.191	0.137	0.183	0.186	0.020	0.043	0.091
	840.58	1170.8	2565.2	777.91	962.09	2147.7	824.31	1223.5	2758.2	535.86	562.68	1350.1

$d(N) = \sqrt{N}$ in (a), the ratio is 0.0009, which means only $0.0009 \times 2^{25} = 30,198$ information criterion computations were executed out of $2^{25} = 33,554,432$.

For small p, the ration is around 10%, which means that Algorithm 1 makes ten times faster than the same computation without B&B. However, it is not so efficient compared with for larger p values. However, the computation does not take so much time either even if we do not use B&B. For large p, we observe that the computation is extremely efficient using Algorithm 1, which is the most attractive feature.

In particular, we observe that for the standard data, the proposed procedure runs more than ten times faster and more than 100 times faster than the original without using B&B, which is surprising because B&B runs at most three to five times for discrete data (Suzuki and Kawahara, 2017). Another significant observation is that the efficiency does not decay so much even when $d(N)$ is small while AIC ($d(N) = 1$) does not work efficiently for the discrete B&B procedures.

We also analyzed when Algorithm 1 does not work efficiently. If we compare (a)(b)(c) in Table 1, we find that (c) is the least efficient among the three. Note that the correlation between X_p and other $p-1$ variables is the least in (b), and that the correlation among the $p-1$ variables is the least in (c). Although the efficiencies for (a) and (b) are almost similar, those for (a)(b) and (c) are significantly different. If we regard X_p and other $p-1$ variables as a response and predictors, respectively, we can consider that colinearlity among the $p-1$ variables affects the performance: if the colinearlity is large, since some redundant variables are included, Algorithm 1 detects unnecessary variables and save computations. However, if the correlation among them is small, Algorithm 1 needs all the computations.

4.2 Using Actual Datasets

We apply Algorithm 1 to datasets Hitters and breastcancer in the R packages ISLR and gRbase, respectively. Hitters ($N = 322$ and $p = 20$) contains 59 missing values and three catagory data, so that we removed them ($N = 263$ and $p = 17$). The 17 variables are "AtBat", "Hits", "HmRun", "Runs", "RBI", "Walks", "Years","CAtBat","CHits","CHmRun","CRuns","CRBI","CWalks", "PutOuts", "Assists", "Errors","Salary". The breastcancer data set consists of $N = 250$ samples for 1000 continuos variables (gene expression) and one binary variable (case/control), and we use the first 20 gene expression data ($N = 250$, $p = 20$).

For each of the p variables $X_1, \cdots, X_p$, we computes a parent set $\pi \subset S$ of X_i for each $S \subseteq \{1, \cdots, p\} \backslash \{i\}$ and $i = 1, \cdots, p$, where $p = 17$ and $p = 20$ for the Hitters and breastcancer data sets.

We show the actual execution times in Figure 2 for the two data sets, and see that more than five times faster and more than 20 times faster than the original procedure (without B&B).

5. Concluding Remarks

We constructed a dynamic programming framework and proposed a B&B approach for continuous BNSL. The current bound works only for information criteria with large $d(N)$. However, those $d(N)$ satisfy (2), and we may use those information criteria in various applications. In fact, in this case, the sizes of parent sets are small, and the resulting Bayesian network is sparse (has few edges).

Intuitively, it seems that the pruning is easier for discrete BNSL than for continuous BNSL. This is because the penality term increases by only one each time a variable is added to the parent set for the continuous variable while the number of variables is multiplied at least two for discrete BNSL.

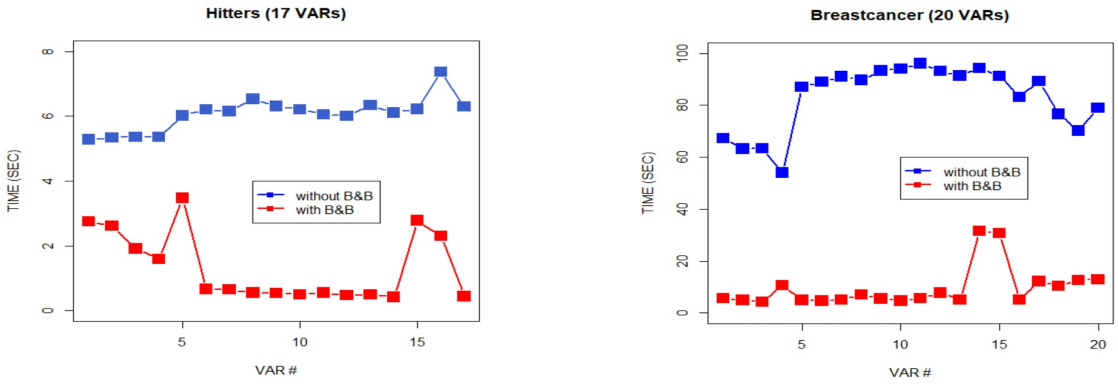


Figure 2:

Surprizingly, we find that the continuous counterpart is much more efficient. We are not sure about the exact reason but the lower-bounds are different between discrete and continuous B&B.

Future work includes finding a better lower bound and finding the exact conditions under which bound works best.

References

H. Akaike. Information theory and an extension of the maximum likelihood principle. In *2nd International Symposium on Information Theory*, volume 57, Budapest, Hungary, 1973.

W. Buntine. Theory refinement on Bayesian networks. In *Uncertainty in Artificial Intelligence*, pages 52–60, Los Angels, CA, 1991.

C. P. Campos and Q. Ji. Efficient structure learning of Bayesian networks using constraints. *Journal of Machine Learning Research*, 12:663–689, 3 2011.

D. M. Chickering, C. Meek, and D. Heckerman. Large-sample learning of Bayesian networks is NP-hard. In *Uncertainty in Artificial Intelligence*, pages 124–133, Acapulco, Mexico, 2003. Morgan Kaufmann.

G. Cooper and E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9(4):309–347, 1992.

J. Cussens and M. Bartlett. *GOBNILP 1.6.2 User/Developer Manual1*. University of York, 2015.

D. Eddelbuettel. *Seamless R and C++ Integration with Rcpp*. Springer-Verlag, 2013.

E. J. Hannan and B. G. Quinn. The determination of the order of an autoregression. *Journal of the Royal Statistical Society, Series B*, 41:190–195, 1979.

S. Ott, S. Imoto, and S. Miyano. Finding optimal models for small gene networks. In *9Th Pacific Symposium on Biocomputing*, pages 557–567, 2004.

J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference (Representation and Reasoning)*. Morgan Kaufmann, 2nd edition, 1988.

J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.

G. E. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6(2):461–464, 1978.

T. Silander and P. Myllymaki. A simple approach for finding the globally optimal Bayesian network structure. In *Uncertainty in Artificial Intelligence*, pages 445–452, Arlington, Virginia, 2006. Morgan Kaufmann.

A. P. Singh and A. W. Moore. Finding optimal Bayesian networks by dynamic programming. Technical report, Carnegie Mellon University, 2005.

P. Spirtes, C. Glymour, and R. Scheines. *Causation,Prediction and Search*. Springer Verlag, Berlin, 1993.

J. Suzuki. A construction of Bayesian networks from databases based on an mdl principle. In *Uncertainty in Artificial Intelligence*, pages 266–273, Washington DC, 1993. Morgan Kaufmann.

J. Suzuki. Learning Bayesian belief networks based on the minimum description length principle: An efficient algorithm using the b & b technique. In *International Conference on Machine Learning*, pages 462–470, Bari, Italy, 1996. Morgan Kaufmann.

J. Suzuki. On strong consistency of model selection in classification. *IEEE Trans. on Information Theory*, IT-52(11):4766–4774, 11 2006.

J. Suzuki. An efficient Bayesian network structure learning strategy. *Next Generation Computation Journal*, 1, 2016.

J. Suzuki. A theoretical analysis of the BDeu scores in Bayesian network structure learning. *Behaviormetrika*, 1:1–20, 2017.

J. Suzuki and J. Kawahara. Branch and bound for regular Bayesian network structure learning. In *Uncertainty in Artificial Intelligence*, pages 212–221, Sydney, 2017.

J. Tian. A branch-and-bound algorithm for MDL learning Bayesian networks. In *Uncertainty in Artificial Intelligence*, pages 580–588, Stanford, CA, 2000. Morgan Kaufmann.

M. Ueno. Learning likelihood-equivalence Bayesian networks using an empirical Bayesian approach. *Behaviormetrika*, 35(2):115–135, 2008.

C. Yuan and B. Malone. Learning optimal bayesian networks: A shortest path perspective. *Journal of Artificial Intelligence Research archive*, 48(1):23–65, 2013.

List of Authors