Journal of Physics: Conference Series

PAPER • OPEN ACCESS

Exact Bayesian learning of Partition Directed Acyclic Graphs

To cite this article: J Pensar *et al* 2018 *J. Phys.: Conf. Ser.* **1036** 012017

Exact Bayesian learning of Partition Directed Acyclic Graphs

J Pensar[1,5] and J Kohonen[2,5] and J Corander[1,3,4]

[1]Department of Mathematics and Statistics, University of Helsinki, Finland
[2]Department of Computer Science, University of Helsinki, Finland
[3]Department of Biostatistics, University of Oslo, Norway
[4]Wellcome Trust Sanger Institute, Cambridge, UK

E-mail: jukka.corander@medisin.uio.no

Abstract. It is known that directed acyclic graphs (DAGs) may hide several local features of the joint probability distribution that can be essential for some applications. To remedy this, more expressive model classes have been introduced. In addition to the restrictions implied by conditional independence, these model classes typically include some form of local structure that implies equality constraints on the node-wise conditional distribution. In particular, the concept of context-specific independence (CSI) was introduced to increase the flexibility of traditional Bayesian networks. Furthermore, in the most expressive class of generalized Bayesian networks, decision graphs were used to model arbitrary parameter restrictions. Here we formulate an alternative representation of such models called a partition DAG (PDAG), which defines the parameter restrictions using a partition-based representation of the parent outcome spaces. We establish a criterion that can identify whether an arbitrary PDAG has a CSI-consistent representation using an efficient basic graph theoretic algorithm. Based on a recursive inference algorithm for partition posteriors, an exact Bayesian learning method is introduced. We demonstrate on real data that exact learning of PDAGs can identify important relationships between variables that have not been discovered by previous graphical model learning methods.

1. Introduction

Bayesian networks have represented a standard workhorse in artificial intelligence and machine learning for more than two decades [15]. However, it has also been acknowledged that the directed acyclic graphs (DAGs) they are based upon, may hide several local features of the joint probability distribution that can be essential for some applications. To remedy this, more expressive classes of Bayesian networks have been introduced [4, 5, 9, 10, 17, 18, 20]. In particular, several of the introduced model classes [4, 9, 17, 20] have been built around the concept of context-specific independence (CSI), which is a natural generalization of conditional independence. In addition to generalized Bayesian networks, the related classes of probabilistic decision graphs (PDGs) [12] and chain event graphs (CEGs) [22] have been introduced for the purpose of capturing asymmetric model structures.

The common feature among most classes of generalized Bayesian networks is that additional local restrictions impose some form of equality constraints on the node-wise conditional

[5] JP and JK contributed equally to this work.

distributions. The equality constraints effectively partition the outcome space of the parents of a node into classes, such that parent configurations belonging to the same class induce the same distribution. The structure of the possible partitions depends on the generality of the restrictions [18]. In this work we will consider Bayesian networks with arbitrary equality constraints, that is, arbitrary partitions. This corresponds to using decision graphs for modeling the local structure between a node and its parents [5], however, we will consider an alternative representation based on the actual parent outcome partitions. In contrast to decision graphs, each distinct partition will correspond to a unique set of restrictions. We refer to the model structure containing both the global DAG structure and the local parent outcome partitions as a partition DAG (PDAG). When each partition in a PDAG has maximum cardinality, only the independence characteristics of the DAG itself are retained. With a decreasing number of classes in a partition, an increasing set of additional local constraints are introduced to the conditional distribution of the corresponding node. Some of these will typically correspond to CSIs, but in general not all of them will be interpretable as such.

The modular composition of Bayesian networks is a very important structural property in terms of model learning. In particular, for a given a DAG, the variables specifying the local structure of a node are confined to the parents of the node, enabling efficient learning of Bayesian networks with local structure. In contrast, one of the key features of models such as PDGs and CEGs is the ability to model conditioning contexts that consist of specific joint configurations of large collections of variables. Although these high-order interactions make the models extremely flexible, they also make learning of such graphs very challenging. As a result, methods developed for PDG and CEG learning have exploited Bayesian network learning as a step of the learning procedure [13, 1]. Still, numerical experiments have indicated the difficulty in learning PDG/CEG models with higher predictive accuracy than standard Bayesian networks [13, 21].

The paper is organized as follows. In Section 2 we introduce PDAGs and discuss their connection to the well-known concept of CSI. In section 3 we introduce a graph theoretical condition for determining if an arbitrary PDAG has a CSI-based dependence structure, and if not, we explain how one can modify the partitions in order to make them consistent with CSI. Section 4 describes how to evaluate the Bayesian score of a PDAG for a given set of data. Section 5 presents a recent recursive algorithm for exact learning of the optimal parent partitions. The algorithm can in the binary case efficiently identify the maximum *a posteriori* partition of a parent outcome space for up to four binary parents. In Section 6 we demonstrate using real data that exact learning of PDAGs can identify important relationships between variables that have not been earlier discovered by any other graphical model learning methods. Finally, in Section 7 we provide some additional remarks and discuss some ideas for future research.

2. Partition Directed Acyclic Graphs

A DAG is a graph $G = (V, E)$ consisting of a set of nodes $V = \{1, \ldots, d\}$ and a set of edges $E \subset V \times V$ such that $(u, v) \in E$ if there is a directed edge from node u to v. The acyclicity restriction prevents the edge set from containing directed cycles. The set of parents of node v is denoted by $pa(v) = \{u \in V : (u, v) \in E\}$. In a Bayesian network, the nodes V correspond to a set of stochastic variables $X = \{X_1, \ldots, X_d\}$. We use X_S, where $S \subseteq V$, to denote a set of variables and we use lower case letters x_S to denote a value taken by X_S. The outcome space of a set of variables X_S is denoted by $\mathcal{X}_S$, and the cardinality is denoted by $|\mathcal{X}_S|$. In this work, we assume that all considered variables are binary, $\mathcal{X}_v = \{0, 1\}$. Our results generalize to non-binary variables in a relatively straightforward manner.

A Bayesian network models the joint distribution over X_V by asserting statements of conditional independence,

$$X_A \perp X_B \mid X_S \Leftrightarrow p(X_A \mid X_B, X_S) = p(X_A \mid X_S).$$

X_1	X_2	X_3	$p(X_4 \mid X_{1,2,3})$
0	0	0	p_1
0	0	1	p_2
0	1	0	p_3
0	1	1	p_4
1	0	0	p_5
1	0	1	p_6
1	1	0	p_7
1	1	1	p_8

(a)

X_1	X_2	X_3	$p(X_4 \mid X_{1,2,3})$
0	0	0	p_1
0	0	1	p_2
0	1	0	p_3
0	1	1	p_3
1	0	0	p_4
1	0	1	p_5
1	1	0	p_3
1	1	1	p_6

(b)

X_1	X_2	X_3	$p(X_4 \mid X_{1,2,3})$
0	0	0	p_1
0	0	1	p_2
0	1	0	p_2
0	1	1	p_3
1	0	0	p_4
1	0	1	p_2
1	1	0	p_2
1	1	1	p_5

(c)

Table 1. CPT of node 4 when $pa(4) = \{1, 2, 3\}$: (a) no regularities, (b) CSI-based regularities, and (c) arbitrary regularities.

The conditional independence statements assumed by a Bayesian network are compactly encoded by its DAG, where edges correspond to direct dependencies and lack of edges corresponds to conditional independencies. More specifically, the dependence structure is characterized by the local directed Markov property, which implies an explicit factorization of the joint distribution according to

$$p(X_1, \ldots, X_d) = \prod_{v=1}^{d} p(X_v \mid X_{pa(v)}). \tag{1}$$

In (1), each factor corresponds to a collection of conditional probability distributions (CPDs). By definition, each collection of CPDs implicitly assumes full dependence of v on its parents, i.e. separate parameters are needed to specify the CPD for all combinations in $\mathcal{X}_{pa(v)}$. In general, this requires

$$(|\mathcal{X}_v| - 1) \cdot \prod_{u \in pa(v)} |\mathcal{X}_u|$$

free parameters. In other words, under the assumption of binary variables, the number of parameters needed to specify all CPDs over X_v is $2^{|pa(v)|}$. The standard approach for representing the CPDs associated with a node is to use a conditional probability table (CPT). An example of a traditional CPT with distinct CPDs is shown in Table 1(a).

A limitation of traditional Bayesian networks is that the number of model parameters associated with a node grows exponentially with the number of parents. However, in many situations, there may exist regularities in form of identical CPDs within a CPT. One of the earliest and most well-known approaches for capturing such regularities introduced the concept of context-specific independence (CSI) [4]:

$$X_A \perp X_B \mid x_C, X_S \Leftrightarrow p(X_A \mid X_B, x_C, X_S) = p(X_A \mid x_C, X_S).$$

CSI is a natural generalization of conditional independence that only holds in part of the outcome space, as specified by the context $X_C = x_C$. In terms of CPTs, we consider local CSIs of the form

$$X_v \perp X_u \mid x_{pa(v)\setminus u}, \text{ where } u \in pa(v),$$

since such statements will imply identical CPDs within the CPT:

$$p(X_v \mid x_u, x_{pa(v)\setminus u}) = p(X_v \mid x'_u, x_{pa(v)\setminus u}) \text{ for all } x_u, x'_u \in \mathcal{X}_u.$$

In the binary case, a local CSI thus corresponds to a restriction forcing exactly two values in the CPT of v to be identical. As an example, consider the CPT in Table 1(b), where the parent configurations $(0,1,0),(0,1,1),(1,1,0)$ all induce the same conditional probability p_3. A closer examination reveals that this particular CPT structure can be explained by the CSIs

$$X_3 \perp X_4 \mid X_1 = 0, X_2 = 1 \text{ and } X_1 \perp X_4 \mid X_2 = 1, X_3 = 0.$$

Rather than specifying the same distribution multiple times, the above parent configuration can then be merged into a single class of configurations. Hence, the set of restrictions imposed by a collection of CSIs defines a partition of the parent outcome space into classes, where all elements in the same class induce an identical CPD.

Note, however, that the converse is not true, that is, an arbitrary partition of the parent outcome space does not necessarily correspond to a collection of CSIs. In this work, we go beyond CSI-based regularities and consider arbitrary regularities of the form

$$p(X_v \mid x_{pa(v)}) = p(X_v \mid x'_{pa(v)}).$$

As an example, consider the CPT in Table 1(c), where $(0,0,1),(0,1,0),(1,0,1),(1,1,0)$ all induce the same conditional probability p_2. In contrast to the previous example (Table 1(b)), these regularities cannot be fully explained by CSI and we need to allow arbitrary partitions to compactly represent the CPT structure. We discuss this more in detail in the next section, where we also introduce a criterion for determining whether the regularities in a CPT can be explained by CSI or not.

In general, Bayesian networks with parent outcome partitions enjoy similar flexibility in terms of parameter restrictions as probabilistic decision graphs, but are based on a different representation that enables efficient learning algorithms to be developed. To formally represent the structure of Bayesian networks with arbitrary parent outcome partitions, we introduce the concept of partition DAGs.

Definition 1. *Partition DAG*
Let $G = (V, E)$ be a DAG for the stochastic variables $\{X_1, \ldots, X_d\}$. For all $v \in V$, let S_v be a partition of the set $\mathcal{X}_{pa(v)}$ into k classes $s_1, \ldots, s_k$ such that the conditional probabilities $p(x_v \mid x_{pa(v)})$ are equal for all $x_{pa(v)} \in s_c, c = 1, \ldots, k$ and unrestricted otherwise. The graph G together with the collection of partitions $\mathcal{S} = \{S_v : v \in V\}$ define a Partition DAG (PDAG), denoted by $G_{\mathcal{S}}$.

For notational simplicity, indexing of the partition S_v and the number of classes k in it with respect to the node is not explicitly shown for each class of S_v when the meaning is unambiguous. Note that k may freely vary across the nodes $v \in V$.

3. CSI-consistent partition

In [17], a parent outcome partition is referred to as CSI-consistent if it can be constructed according to a collection of local CSIs. As discussed in the previous section, an arbitrary partition would typically be expected to reside beyond the scope of CSI, however, then the following questions arise: what restrictions must a partition satisfy in order to be CSI-consistent, and if a partition is not CSI-consistent, how can it be modified to become CSI-consistent? CSI-consistency can be a useful property. For example, previous research has shown that CSI can be exploited to improve the efficiency of inference algorithms [19, 20].

To construct a procedure for determining if a partition is CSI-consistent we introduce the following graphical criterion.

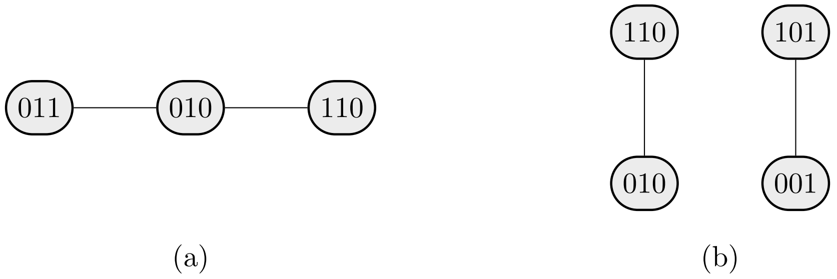

Figure 1. The class connection graph of the non-singleton class in the partition obtained from the CPT in (a) Table 1(b) and (b) Table 1(c). For clarity, the nodes are labeled with the corresponding parent configurations.

Definition 2. *Class connection graph*
A class connection graph $G_c = (N, E)$ for a class c for $v \in V$ is an undirected graph over the class elements $s_c = \{x^{(l)}_{pa(v)}\}^q_{l=1}$ such that $q = |s_c|$ and

$$N = s_c \text{ and } \{l, r\} \in E \text{ if } d(x^{(l)}_{pa(v)}, x^{(r)}_{pa(v)}) = 1,$$

where $d(\cdot, \cdot)$ is the Hamming distance between the two configurations.

Theorem 1. *A partition $S_v = \{s_1, \ldots, s_k\}$ is CSI-consistent if and only if the corresponding class connection graphs $G_1, \ldots, G_k$ are connected.*

Proof. See Appendix A. □

Theorem 1 basically reformulates the problem of checking for CSI-consistency to a well-known graph connectivity problem for which there already exist several efficient methods such as the depth-first and breadth-first algorithms [11].

To illustrate the criterion, consider the partition obtained from the CPT in Table 1(b). The class connection graph of the non-singleton class is shown in Figure 1(a). Note that the class connection graph of a singleton class is connected by definition. Since the graph in Figure 1(a) is connected, the partition is CSI-consistent. On the other hand, consider the partition obtained from the CPT in Table 1(c). The class connection graph of the non-singleton class is shown in Figure 1(b). In this case, the graph is not connected and, consequently, the partition is not CSI-consistent. However, if we would split the class

$$\{(0, 0, 1), (0, 1, 0), (1, 0, 1), (1, 1, 0)\}$$

into two new classes

$$\{(0, 0, 1), (1, 0, 1)\} \text{ and } \{(0, 1, 0), (1, 1, 0)\},$$

the partition would become CSI-consistent, since all class connection graphs would then be connected. The new representation would no longer be minimal in terms of the number of parameters, since the same distribution would have to be defined twice. Still, the criterion provides a straightforward way of making an arbitrary partition CSI-consistent by removing as few equality constraints as possible.

4. Bayesian score for PDAGs

A very useful property of PDAGs and similar models is that the marginal likelihood can still be evaluated by a closed-form expression [5]. Let $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{n}$ denote a set of training data consisting of n observations $\mathbf{x}_i = (x_{i1}, \dots x_{id})$, assuming that $\mathbf{X}$ contains no missing values. We let θ_{vc} denote the probability of $X_v = 1$ given that $X_{pa(v)}$ is assigned a value in class c of the partition S_v. For a given PDAG $G_{\mathcal{S}}$, we let $\theta_{G_{\mathcal{S}}}$ denote collectively all the unknown parameters in the CPTs. The likelihood of a PDAG can then be written as

$$p(\mathbf{X} \mid \theta_{G_{\mathcal{S}}}, G_{\mathcal{S}}) = \prod_{v=1}^{d} \prod_{c=1}^{k_v} \theta_{vc}^{n(v,c,1)} (1 - \theta_{vc})^{n(v,c,0)},$$

where $n(v, c, 1)$ denotes the total count of observations with $x_v = 1$ and $x_{pa(v)}$ falling into the class c of S_v. Further, $n(v, c, 0)$ is the corresponding count with $x_v = 0$ and $n(v, c) = n(v, c, 0) + n(v, c, 1)$. We use a prior for $\theta_{G_{\mathcal{S}}}$ which factorizes over the DAG and the classes, such that the marginal likelihood (or evidence) of the data can be calculated analytically. Under the standard conjugate Beta prior, we obtain the following expression for the marginal likelihood given the PDAG:

$$p(\mathbf{X} \mid G_{\mathcal{S}}) = \prod_{v=1}^{d} \prod_{c=1}^{k_v} \frac{\Gamma(\alpha)}{\Gamma(n(v,c) + \alpha)} \prod_{l=0}^{1} \frac{\Gamma(n(v,c,l) + \alpha_l)}{\Gamma(\alpha_l)}, \tag{2}$$

where $\alpha = \alpha_0 + \alpha_1$.

There are several options in terms of specifying the hyperparameters in (2), see e.g. [15]. In all the numerical experiments reported here we have used the uniform prior with $\alpha_1 = \alpha_0 = 1$. Following the Bayesian approach of structure learning, we aim at maximizing $\log p(\mathbf{X} \mid G_{\mathcal{S}}) + \log p(G_{\mathcal{S}})$, where $p(G_{\mathcal{S}})$ is a product prior over the d partitions. Each prior factor in the product is assumed to be defined by the uniform distribution on the number of classes k_v in the partition S_v. Further discussion about the choice of priors is found in the final section of the paper.

5. Exact PDAG learning algorithm

Learning PDAGs from training data is in general extremely challenging. Even for a fixed ordering of the variables specified by a candidate DAG, there exist extremely many possible partitions of the parent outcome space for each single node. For example, for a node with 3 (binary) parents, there are 4140 ways to partition the 8 possible parent configurations, whereas for a node with 4 parents, the corresponding number is already 10,480,142,147. In general, the number of unordered partitions of $|\mathcal{X}_{pa(v)}|$ entries is given by the Bell number of $|\mathcal{X}_{pa(v)}|$. While direct complete enumeration of all possible partitions is in practice tedious for cases with more than two parents, the recursive algorithm for exact posterior inference about clustering introduced in [14] is sufficiently fast for exact learning of partitions in a PDAG with at most 4 parents for any node.

Let $S(\mathcal{X}_{pa(v)}, k_v)$ be the Stirling number of the second kind, i.e. the number of possible partitions of $\mathcal{X}_{pa(v)}$ into exactly k_v non-overlapping and non-empty subsets. We set the prior probability of a partition S_v equal to $|\mathcal{X}_{pa(v)}|^{-1} S(\mathcal{X}_{pa(v)}, k_v)^{-1}$, which corresponds to the uniform distribution for the number of classes k_v for node v. To enable use of the recursive algorithm for posterior inferences, we consider now *ordered partitions*, each of which is a tuple of disjoint nonempty subsets, whose union is $\mathcal{X}_{pa(v)}$.

The posterior probability of an ordered partition S_v with k_v classes in the PDAG model equals

$$p(S_v \mid \mathbf{X}) = \frac{p(S_v) p(\mathbf{X} \mid S)}{p(\mathbf{X})} = Z \cdot \prod_{c=1}^{k_v} f(s_{v,c}),$$

where Z is the normalizing constant and $f(s_{v,c})$ is the marginal likelihood term for class c in (2). The posterior probability of k clusters equals under the above formulation

$$p(k_v \mid \mathbf{X}) = Z \cdot \sum_{\substack{S_v \in \mathcal{S}_v \\ |S_v|=k}} \prod_{c=1}^{k_v} f(s_{v,c}), \tag{3}$$

where $\mathcal{S}_v$ is the space of ordered partitions for node v. This sum of products can be conveniently expressed by subset convolution. Given two real-valued functions f and g defined on the subsets of $\mathcal{X}_{\Pi_v}$, their *subset convolution* is the function

$$(f * g)(Y) = \sum_{A \subseteq Y} f(A) \cdot g(Y \setminus A), \quad \text{for all } Y \subseteq \mathcal{X}_{pa(v)}. \tag{4}$$

Expressed in a more symmetric form we get

$$(f * g)(Y) = \sum_{\substack{A,B \subseteq X \\ A+B=Y}} f(A) \cdot g(B),$$

where $A+B=Y$ represents disjoint union. Convolution is associative, and iterative application hence yields

$$(f_1 * \ldots * f_{k_v})(Y) = \sum_{\substack{A_1,\ldots,A_{k_v} \subseteq Y \\ A_1+\ldots+A_{k_v}=Y}} \prod_{c=1}^{k_v} f_c(A_c).$$

The k_v-fold convolution expresses summation over ordered k_v-partitions of a set Y. Writing (3) in terms of iterated convolution yields

$$p(k_v \mid \mathbf{X}) = Z \cdot f^{(k_v)}(\mathcal{X}_{pa(v)}), \tag{5}$$

where $f^{(k_v)} = (f * \ldots * f)$ denotes the convolution of k_v copies of f.

The full convolution table for $f*g$ can be obtained in $O(3^{|\mathcal{X}_{pa(v)}|})$ operations, which is still very fast for the case with 4 parent nodes. In practice these calculations can be done in approximately 1 minute for such a node of a PDAG model, whereas 3 parent nodes can be handled in a fraction of a second. When the summation in (3) is replaced with maximization, one obtains the maximum posterior probability among k_v-partitions. This can be computed using a variant of subset convolution, where the summation is replaced with maximization such that the subset convolution is performed over the max-product semiring, instead of the sum-product ring. This yields an $O(|\mathcal{X}_{pa(v)}|3^{|\mathcal{X}_{pa(v)}|})$ algorithm for finding the maximum *a posteriori* (MAP) partition for any node-parent combination in a PDAG.

6. Experiments

6.1. Heart disease data

A classical data set used for benchmarking various graph learning algorithms is the coronary heart disease risk factor data set with 1841 observations on 6 binary variables [8, 23, 17]. Definitions of the variables are given in Table 2. The data set has been considered also in numerous other articles about graphical model learning and a consistent finding is that the Family anamnesis of coronary heart disease is independent of all the remaining variables.

Figure 2 shows the DAG with the parent sets of nodes being determined by exact learning such that at most four parent candidates were considered simultaneously and the available partial

Variable	Outcomes
X_1: Smoking	$\text{No} := 0, \text{Yes} := 1$
X_2: Strenuous mental work	$\text{No} := 0, \text{Yes} := 1$
X_3: Strenuous physical work	$\text{No} := 0, \text{Yes} := 1$
X_4: Systolic blood pressure	$< 140 := 0, > 140 := 1$
X_5: Ratio of β and α lipoproteins	$< 3 := 0, > 3 := 1$
X_6: Family anamnesis of CHD	$\text{No} := 0, \text{Yes} := 1$

Table 2. Description of the variables in coronary heart disease (CHD) data.

X_{1356}	#0	#1	$\bar{X}_4$
0001	26	15	0.3659
1000	107	45	0.2961
1100	168	76	0.3115
class 1	301	136	0.3112
class 2	753	651	0.4637

(a)

X_{1256}	#0	#1	$\bar{X}_4$
0011	16	9	0.3600
1000	185	79	0.2992
1100	90	42	0.3182
class 1	291	130	0.3088
class 2	763	657	0.4627

(b)

X_{156}	#0	#1	$\bar{X}_4$
100	275	121	0.3056
class 1	275	121	0.3056
class 2	779	666	0.4609

(c)

Table 3. MAP partition for X_4 with parent set (a) 1356, (b) 1256, and (c) 156. Details of class 2 have been omitted. For each combination of parent values, #0 and #1 are the counts of $X_4 = 0$ and $X_4 = 1$, respectively, and the last column is the proportion of ones.

ordering of the variables was used to exclude certain candidate nodes from the set of possible parents. In practice, nodes 4 and 5 had each four possible parents, whereas node 3 had two and node 2 only one. Nodes 1 and 6 had no possible parents given the imposed partial ordering information. For a detailed discussion of these data see, e.g. [23].

In the more comprehensive analysis reported below, we concentrate on variable X_4 (systolic blood pressure) and its parent structure. As at most four parent candidates were considered at a time, there are $\binom{5}{4} = 5$ different candidate sets of parents, in shorthand notation: 2356, 1356, 1256, 1236 and 1235.

For each candidate set we obtained the MAP partition for the 16 parent value combinations as discussed before. The log marginal likelihoods of the MAP partitions (relative to the trivial partition) are +0.0, +13.5, +13.4, +0.0 and +10.5, respectively. The two highest likelihoods (with almost a tie) occur for the candidate sets 1356 and 1256, which are highly similar to each other. In both cases the MAP partition is a 2-partition where the smaller class contains three parent value combinations, as illustrated in Tables 3(a)–(b).

In both tables, we can see that the smaller class contains parent configurations, for which X_4 has a low probability of high blood pressure. In both cases, the first row, which contains relatively few observations overall, seems somewhat odd and might be a random artefact. The remaining two rows, 1000 and 1100 in either case, appear more plausible, and they would seem to indicate an interesting phenomenon in the data: if $(X_1, X_5, X_6) = (1, 0, 0)$, then the probability of $X_4 = 1$ is much lower than otherwise.

This dependence, where just one particular combination of three binary variables has a joint effect on a fourth variable, is not explicitly caught by the structure of ordinary DAG models. As mentioned above, despite of numerous previous analyses of this data with various Bayesian and Markov network models, any significant association between the family anamnesis of CHD

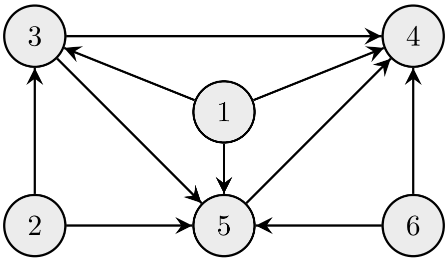

Figure 2. DAG structure for the heart disease data.

and the remaining variables has never been detected. However, it has been remarked that the lack of an association between node 6 and the two biologically measured variables (nodes 4 and 5) is surprising, given that CHD among relatives is a known risk factor.

To explore the association a bit further, we also examine the optimal parent set with only 3 nodes. This contains the nodes 156 and the MAP partition of the 8 parent configurations is shown in Table 3(c). Again, we obtain a 2-partition, but now the smaller class contains just the one parent configuration 100, while the seven other combinations are in the other class. The finding can be interpreted such that individuals that smoke and have a low lipoprotein ratio as well as no family anamnesis of CHD, have a lower probability of high blood pressure than others. The reason why the connection between the family history variable (node 6) and the other CHD risk factors is not revealed in ordinary pairwise association or network analysis, is the context-specific nature of their dependence, which is masked by the distribution of the other conditional probabilities in Table 3(c). All these probabilities are within 3 percentage points from the mean of class 2 (0.46), whereas only the configuration 100 corresponds to much smaller probability (0.31), still based on substantial amount of data (396 observations).

6.2. Simulated data

To gauge the behavior of the learning algorithm, we performed a simulated data study using the CSI-based model with 10 binary variables introduced in [17]. Due to space limitations, we focus here on variable X_5, which has three parents X_1, X_4 and X_8, and satisfies the CSI

$$X_5 \perp X_4, X_8 \mid X_1 = 0.$$

In other words, there are 8 parent configurations which are partitioned into 5 classes such that one class contains the four configurations where $X_1 = 0$, and the four remaining classes contain a single configuration.

In the simulation, we generated 100 random replicates of data with varying sizes according to the given model, and then identified the MAP partition using the subset convolution algorithm for each replicate. The MAP partition was compared to the generating partition using two metrics: adjusted Rand index, and zero-one-loss (one if the partitions are identical, zero otherwise). The results of both metrics are shown in Figure 3.

As expected, the more data we have, the closer the learned partition is to the structure in the generating model. We note that fairly large sample sizes are required to obtain very accurate results. This is reasonable, since the PDAG model class is extremely flexible and e.g. 1000 observations split over the 16 value combinations (two child node outcomes for each combination of parental values) means on average only 60 samples per category and the variation around this may also be substantial. It is important to notice the difference between the generating distribution used for the simulations and the structure of the real CHD data analyzed previously. As shown for the CHD data, the conditional probabilities were highly similar across

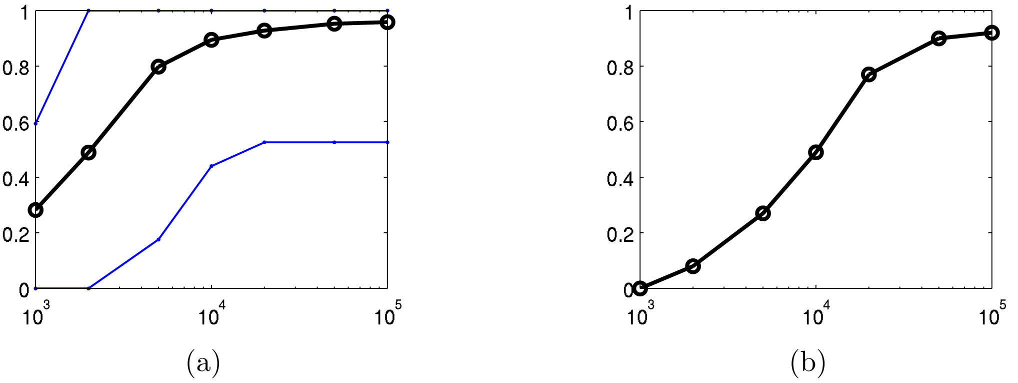

Figure 3. Simulated data study with data sizes ranging from 1000 to 100000. Vertical axis: (a) adjusted Rand index between inferred and the generating partition (black: average over 100 replications; blue: 5% and 95% quantiles), (b) proportion of inferred partitions equal to the structure of the generating model in 100 replications.

the parental combinations of node 4, except in one case. In contrast, the conditional probabilities in the synthetic model have been sampled from a prior distribution (when not forced equal), which means that they are more uniformly spread across the interval $(0, 1)$, such that any two probabilities may coincidentally be fairly close to each other. This would typically require more data for accurate learning of the underlying partition.

7. Discussion

Our approach has focused on the representability and interpretability of partitions of CPT elements to encode local restrictions on the parameters of a DAG model. The exact Bayesian learning method we introduced uses recursive computation to identify the posterior optimal partition for a node given any candidate set of parents. However, direct brute-force approach still necessitates enumeration across such candidate sets, which is in general infeasible. It would therefore be interesting to explore how the recursive computation over subsets of parental outcomes could be combined with an approach based on computational logic to identify the optimal parent sets. There has recently been a considerable interest in use of computational logic algorithms for learning DAGs and Markov networks, such as integer programming, answer-set programming and maximum satisfiability [7, 2, 6, 3, 16]. In particular, the maximum satisfiability approach introduced in [3] scales surprisingly well with respect to the number of nodes and could potentially leverage a solution to more scalable exact PDAG learning in combination with the subset convolution.

The prior we use for partitions of CPT elements is a uniform distribution for the number of classes in the partition. This prior penalizes most partitions that have cardinality approximately in the middle of the range from a single class to maximum cardinality, as the Stirling number of the second kind is maximized for them on a given set of parent configurations. The rationale for our current prior can be explained as follows. When the number of classes in a partition is small, the model for the conditional distribution of the node is fairly simple as the node is almost contextually independent of its parents except in a limited set of cases. Since there are only few different such models for a candidate set of parents, the prior penalty needs not to be as large. Similarly, if there are many classes, i.e. the partition has almost maximal cardinality, the model is again *structurally* relatively simple since nearly all combinations of the values of parents have their respective unique conditional distributions. When the partition cardinality is neither small nor large, the resulting model has a more diverse set of dependencies and local independencies.

Since there are many such models, some of them may coincidentally provide a fairly good fit to the data, and therefore, the additional penalty from the uniform prior on the number of classes can be a useful way to prevent too flexible models to overfit the data. However, more research is needed to establish the advantages and disadvantages of different types of possible priors for this kind of flexible models.

The partition-based formulation chosen here to represent parameter constraints may also prove particularly useful for inference compared with decision graphs. For instance, [19] concluded that rule-based representations may be more efficient than tree-based and [24] analyzed potential computational advantages resulting from CSIs. Additionally, [20] improved the approach of [19] by using a combination of contexts and tables, leading to contextual belief networks. As noted in [17], the underlying CSIs of a CSI-consistent partition directly correspond to the parent contexts of a contextual belief network. By making the partitions of a PDAG CSI-consistent, the inference method in [20] can thus be applied as such. Whether it is possible to utilize the structure of PDAGs for inference even more efficiently is an open problem.

Acknowledgments
The authors would like to thank professor Teemu Roos for useful discussions on context-specific graphical models.

Appendix A. Proof of Theorem 1

Proof. A partition is CSI-consistent if the structure of all of its classes can be explained by local CSI statements. That is, for each class s_c and each distinct pair of elements $(x_{pa(v)}^{(l_1)}, x_{pa(v)}^{(l_m)})$ within that class, the implied regularity

$$p(x_v \mid x_{pa(v)}^{(l_1)}) = p(x_v \mid x_{pa(v)}^{(l_m)})$$

must be explained by a collection of local CSI statements. This is equivalent to saying that there must exist a sequence of elements $(x_{pa(v)}^{(l_i)})_{i=2}^{m-1}$ in the class s_c for which

$$d(x_{pa(v)}^{(l_i)}, x_{pa(v)}^{(l_{i+1})}) = 1 \quad \text{for } i = 1, \dots, m-1, \tag{A.1}$$

where $d(\cdot,\cdot)$ denotes the Hamming distance. If (A.1) holds, the equality restrictions in the chain

$$p(x_v \mid x_{pa(v)}^{(l_1)}) = p(x_v \mid x_{pa(v)}^{(l_2)}) = \ldots = p(x_v \mid x_{pa(v)}^{(l_{m-1})}) = p(x_v \mid x_{pa(v)}^{(l_m)})$$

can all be explained by local CSIs of the form

$$X_v \perp X_u \mid x_{pa(v)\setminus u}^{(l_i)},$$

where $u \in pa(v)$ is the parent with different values in configurations l_i and l_{i+1}. On the other hand, if

$$d(x_{pa(v)}^{(l_i)}, x_{pa(v)}^{(l_{i+1})}) > 1,$$

the same restriction can no longer be explained be such a statement. It could still be explained by a local CSI of the form

$$X_v \perp X_U \mid x_{pa(v)\setminus U},$$

where $U \subset pa(v)$ and $|U| > 1$. However, for such a CSI to be consistent with the class, there must exist a sequence satisfying (A.1) between the considered elements.

The (non-)existence of a sequence satisfying (A.1) between a pair of elements in a class s_c is equivalent to the corresponding node pair being (dis)connected in the class connection graph G_c. For all pairs in the class to be connected, the corresponding class connection graph must be connected. Consequently, all classes in a partition are CSI-consistent if and only if all the corresponding class connection graphs are connected. □

References

[1] L.M. Barclay, J.L. Hutton and J.Q. Smith. Refining a Bayesian Network using a Chain Event Graph. *International Journal of Approximate Reasoning*, 54:1300–1309, 2013.

[2] M. Bartlett and J. Cussens. Advances in Bayesian network learning using integer programming. In *Proceedings of the 29th Conference on Uncertainty in Artificial Intelligence*, 182–191, 2013.

[3] J. Berg, M. Järvisalo and B. Malone. Learning optimal bounded treewidth Bayesian networks via maximum satisfiability. In *Proceedings of the 17th Conference on Artificial Intelligence and Statistics*, 86–95, 2014.

[4] C. Boutilier, N. Friedman, M. Goldszmidt and D. Koller. Context-specific independence in Bayesian networks. In *Proceedings of the Twelfth Annual Conference on Uncertainty in Artificial Intelligence*, 115–123, 1996.

[5] D.M. Chickering, D. Heckerman and C. Meek. A Bayesian approach to learning Bayesian networks with local structure. In *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence*, 1997.

[6] J. Corander, T. Janhunen, J. Rintanen, H. Nyman and J. Pensar. Learning chordal Markov networks by constraint satisfaction. In *Advances in Neural Information Processing Systems 26*, 1349–1357, 2013.

[7] J. Cussens. Bayesian network learning by compiling to weighted MAX-SAT. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 105–112, 2008.

[8] D. Edwards and H. Tomá. A fast procedure for model search in multidimensional contingency tables. *Biometrika*, 72:339–351, 1985.

[9] N. Friedman and M. Goldszmidt. Learning Bayesian networks with local structure. In *Proceedings of the Twelfth Annual Conference on Uncertainty in Artificial Intelligence*, 252–262, 1996.

[10] D. Geiger and D. Heckerman. Knowledge representation and inference in similarity networks and Bayesian multinets. *Artificial Intelligence*, 82:45–74, 1996.

[11] M.C. Golumbic. *Algorithmic Graph Theory and Perfect graphs: Second Edition.* Elsevier, 2004.

[12] M. Jaeger. Probabilistic decision graphs: Combining verification and AI techniques for probabilistic inference. In *Proceedings of the first European Workshop on Probabilistic Graphical Models*, 81–88, 2002.

[13] M. Jaeger, J.D. Nielsen and T. Silander. Learning probabilistic decision graphs. *International Journal of Approximate Reasoning*, 42:84–100, 2006.

[14] J. Kohonen and J. Corander. Computing exact clustering posteriors with subset convolution. *Communications in Statistics - Theory and Methods*, 45:3048–3058, 2016.

[15] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques.* MIT Press, 2009.

[16] P. Parviainen, H.S. Farahani and J. Lagergren. Learning bounded tree-width Bayesian networks using integer linear programming. In *Proceedings of the 17th Conference on Artificial Intelligence and Statistics*, 751–759, 2014.

[17] J. Pensar, H. Nyman, T. Koski and J. Corander. Labeled directed acyclic graphs: a generalization of context-specific independence in directed graphical models. *Data Mining and Knowledge Discovery*, 29:503–533, 2015.

[18] J. Pensar, H. Nyman, J. Lintusaari and J. Corander. The role of local partial independence in learning of Bayesian networks. *International Journal of Approximate Reasoning*, 69:91–105, 2016.

[19] D. Poole. Probabilistic partial evaluation: Exploiting rule structure in probabilistic inference. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence*, 1284–1291, 1997.

[20] D. Poole and N.L. Zhang. Exploiting contextual independence in probabilistic inference. *Journal of Artificial Intelligence Research*, 18:263–313, 2003.

[21] T. Silander and T.-Y. Leong A Dynamic Programming Algorithm for Learning Chain Event Graphs. In *Proceedings of Discovery Science: 16th International Conference*, 201–216, 2013.

[22] J.Q. Smith and P.E. Anderson. Conditional independence and chain event graphs. *Artificial Intelligence*, 172(1):42–68, 2008.

[23] J. Whittaker. *Graphical Models in Applied Multivariate Statistics.* Wiley, New York, 1990.

[24] N.L. Zhang and D. Poole. On the role of context-specific independence in probabilistic inference. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, 1288–1293, 1999.