Scoring Bayesian Networks with Informative, Causal and Associative Priors

Giorgos Borboudakis and Ioannis Tsamardinos
Computer Science Department, University of Crete
Institute of Computer Science, FORTH
{borbudak,tsamard}@ics.forth.gr

Abstract

A significant theoretical advantage of search-and-score methods for learning Bayesian Networks is that they can accept informative prior beliefs for each possible network, thus complementing the data. Currently however, there are limited practical ways of assigning priors to each possible network. In this paper, we present a method for assigning priors based on beliefs on the presence or absence of certain paths in the true network. Such beliefs correspond to knowledge about the possible causal and associative relations between a pair of variables X and Y. This type of knowledge naturally arises from prior experimental and observational datasets, among others. We show that incorporating such prior knowledge may not only improve the learning of the direction of the causal relations in the network, but also the learning of the network skeleton. This is particularly the case when sample size is low and thus prior knowledge increases in importance. Our approach is based on converting possibly-incoherent beliefs about marginals to joint distributions of priors by use of optimization theory.

1 Introduction

One theoretical advantage of the search-and-score approach to learning Bayesian Networks [1] versus the constraint-based approach [2] is that the former naturally accepts priors for each network. Since the number of possible networks is super-exponential to the number of variables, in a practical setting one has to assign priors in an implicit way, avoiding enumeration of all structures. For example, one could devise an easily-computable function for the prior given a network. In addition, prior network probabilities have to be assigned so that they reflect our prior knowledge on the domain.

In this paper, we present a method that accepts users' beliefs (probabilities) regarding the possible paths between a set of pairs of variables $\langle X, Y \rangle$. Paths between variables directly correspond to **causal** or **associative** relations, e.g., X causes Y, X and Y do not cause each other but have a common ancestor, or X and Y are statistically associated. For each possible network, the method can efficiently compute its prior corresponding to these input beliefs. It can thus be employed by a search algorithm trying to maximize the score of a network.

Causal knowledge is naturally derived from prior experimental data while associative knowledge from observational data. For example, consider a dataset $\mathcal{D}$ measuring the average amount of exercise per week E, calcium in diet C, occurrence of osteoporosis by 60yrs O and smoking S in a cohort of women. A Bayesian Network could be induced by any appropriate learning method. However, if a prior *experimental study* showed that increasing the amount of exercise, reduces the occurrence of the disease, then the knowledge fact that [E causes (i.e., causally affects) O] with probability p should be incorporated during learning. Similarly, if a prior cohort study (observational study) has shown that smoking correlates with reduced exercising then knowledge [S and E are associated] with probability p' should also be included. The belief strengths p and p' depend on

several factors, such as the statistical power of the study, the p-values, and the quality of the prior studies. Notice that the fact [E causes O] *does not* correspond to the presence of the edge $E \rightarrow O$ in the network: the edge implies a *direct* causal relation while [E causes O] does not depend on the context of modeled variables.

In simulated proof-of-concept experiments we show that the new scoring method can indeed take advantage of prior knowledge. When provided with causal knowledge, it is able to better learn *the orientations* of the edges and the causal relations. For example, let us assume that one learns from the data the Markov-Equivalence class of the Bayesian Networks (called the Partially Directed Acyclic Graph (PDAG) or the essential graph) [2] with the maximum likelihood to be $X - Y - Z$. When given prior knowledge that [X causes Z] with high probability, the network $X \rightarrow Y \rightarrow Z$ obtains a higher a posteriori probability than all other networks in the PDAG. In addition, informative priors can also facilitate learning *the skeleton* of the network; intuitively, prior belief that X and Y are associated tends to induce the true edges that connect the two variables.

One important technical difficulty in the proposed method is that of computing the joint distribution of the input path beliefs, e.g., computing $P(X \text{ causes } Y, Y \text{ causes } Z)$ given $P(X \text{ causes } Y) = 0.8$ and $P(Y \text{ causes } Z) = 0.8$. On one hand, there may be several choices for the joint given the same marginal beliefs. For example, in the above scenario we can infer $P(X \text{ causes } Y, Y \text{ causes } Z) \in [0.6, 1]$. Thus, *path beliefs are inherently dependent*. On the other hand, the beliefs maybe *incoherent* [3], i.e,. not extendable to a joint distribution that satisfies the probability axioms. We present a method that computes a joint distribution of the path properties such that: when the path beliefs are coherent the joint is the closest to uninformative priors; when the input beliefs are incoherent the paths' joint is chosen coherent and induces path probabilities that are the closest to the input beliefs. Once the joint is computed, it can be employed to compute the prior of a network, e.g., the prior of $X \rightarrow Y \rightarrow Z$ is proportional to $P(X \text{ causes } Y, Y \text{ causes } Z)$.

There are currently several other methods that make use of prior knowledge when learning a network, e.g., using knowledge regarding the parameters of the network [4], a causal total order of the variables [1] (i.e. totally ruling out all networks that to not admit the given total order), or the presence or absence of directed edges in the network [5] possibly with beliefs assigned to them [6, 7]. Directed edges correspond to *direct causal relations*, i.e,. relations not mediated by any other variable in the model. Being "direct" depends on the context, i.e., the modeled variables. Such knowledge does not naturally arise from other sources such as past datasets or even expert opinion. Other work represents prior knowledge in the form of a prior Bayesian Network: prior probabilities are assigned based on the distance from this network [8]. Again, it is highly unlikely that such complete prior knowledge is available in a domain to construct this prior network. In general, it can be argued that the type of knowledge the existing methods can incorporate during learning is not in a form that can be easily acquired. As a result, uniform - and thus uninformative - priors are commonly used when learning Bayesian Networks from data. *The problem of incorporating informative priors while learning is listed in the list of open problems in a recent causality editorial* [9]

Prior work that specifically considers the problem of path constraints or beliefs is [10, 11]. The method in [10] assumes one *first learns* a Markov-Equivalence class of Maximal Ancestral Graphs (a generalization of Bayesian Networks that admits hidden variables) [2] from data and *then*, prior knowledge in the form of path constraints is imposed on the graph. In contrast, in this work the network is learnt *with the help of the prior knowledge*. Second, in these works the path priors consist of hard constraints that do not admit degrees of belief. In [11] a method is presented for incorporating beliefs on paths, but relies on computationally expensive Markov Chain - Monte Carlo (MCMC) simulations. However, neither the latter, nor any other method dealing with prior knowledge [6, 7] deals with the issues of dependent, and possibly, incoherent beliefs.

2 Background

We assume the reader's familiarity with Bayesian Networks [12, 13] corresponding learning algorithms and just briefly review the basic concepts. Let $\mathcal{V}$ be a set of n random variables $V_1, \ldots, V_n$. In the rest of the paper, we assume discrete variables but the method applies to any type of variables. A **Bayesian Network** (BN) over $\mathcal{V}$ is a pair $\mathcal{B} = \langle \mathcal{G}_\mathcal{V}, \mathcal{P}_\mathcal{V} \rangle$, where $G_\mathcal{V}$ is a **Directed Acyclic Graph** (DAG) representing conditional independencies between variables $\mathcal{V}$, and $P_\mathcal{V}$ is the joint distribution of $\mathcal{V}$. The graph and distribution must be connected by the equation $P_\mathcal{V} = \prod P(V_i | Pa_\mathcal{G}(V_i))$, where

$Pa_{\mathcal{G}}(V_i)$ are the parents of V_i in $\mathcal{G}$. The above equation is equivalent to what is called the Markov Condition. When the network is fixed in a context we drop the indexes $\mathcal{V}, \mathcal{G}$ from the equations.

The **skeleton** of a Bayesian Network $\mathcal{G}$ is the undirected graph which can be constructed by ignoring the orientations of $\mathcal{G}$. A triple of vertices $\langle X, Y, Z\rangle$ is called a **collider** in $\mathcal{G}$, if $X \rightarrow Y \leftarrow Z$ is in $\mathcal{G}$. A collider $\langle X, Y, Z\rangle$ is **unshielded** if X and Z are not adjacent in $\mathcal{G}$. Two BNs are called **Markov equivalent** if: (a) they have the same skeleton, and (b) they have the same set of unshielded colliders. A **Partially Directed Acyclic Graph** (PDAG) (also known as essential graph) is a graph representing a set of Markov equivalent BNs. It has the same skeleton as all BN representatives and an edge is directed if and only if it is invariant in all BN representatives, and is undirected otherwise. We call a **directed path** from X to Y (**denoted as** $X \Rightarrow Y$) in a graph a sequence of unique edges and nodes in the graph $X \rightarrow V_1 \rightarrow \ldots \rightarrow V_j \rightarrow Y$. We **denote as** $X \Leftrightarrow Y$ the case where there is a distinct node $Z \in \mathcal{V}$ that is a common ancestor of X and Y (i.e., $X \Leftarrow Z \Rightarrow Y$) but neither X is an ancestor of Y nor the reverse. A d-**connecting path** (given the empty set) between X and Y exists if either $X \Rightarrow Y$, $X \Leftarrow Y$, or $X \Leftrightarrow Y$. The absence of a d-connecting path between X and Y is **denoted as** $X \nLeftrightarrow Y$. In the rest of the paper, we assume the Faithfulness Condition [2] that (together with the Markov Condition) implies that *there is a d-connecting path between X and Y, if and only if the two nodes are statistically associated (dependent)*. This assumption is important only when considering associative priors.

Assume we are given a complete multinomial dataset $\mathcal{D}$ over variables $\mathcal{V}$. The probability of a network (or model) $\mathcal{G}$ over $\mathcal{V}$ is $P(G|D) = \frac{P(D|G)\cdot P(G)}{P(D)} \propto P(D|G) \cdot P(G)$. Taking the logarithm of each side we obtain $\log P(G|D) \propto \log P(D|G) + \log P(G)$. The first term is the log likelihood of the data given the graph, while the second the log of the prior of the graph. The graph that maximizes $\log P(G|D)$ also maximizes the right-hand side. Bayesian scoring methods such as BDe, BDeu, [8] and K2 [1] try to approximate the log-likelihood based on different assumptions. Thus, in general all such scoring methods can be decomposed as:

$$Sc(G|D) = Sc(D|G) + Sc(G) \tag{1}$$

When priors are uniform the term $Sc(G)$ can be ignored during maximization. In our setting however, this term may become important.

3 Representing Prior Path Beliefs

For any pair $X, Y \in \mathcal{V}$ we may have a prior belief on the possible paths connecting the two variables in the network. It is important that we devise cases for such paths that are *mutually exclusive* and *allow the representation of common types of causal and associative knowledge*. This is possible as follows: we define the variables $r_{i,j}$ taking values in the set $\{\Rightarrow, \Leftarrow, \Leftrightarrow, \nLeftrightarrow\}$ with the semantics $V_i \Rightarrow V_j$, $V_i \Leftarrow V_j$, $V_i \Leftrightarrow V_j$, and $V_i \nLeftrightarrow V_j$ respectively. When the specific variables V_i, V_j we refer to are not important we will use a single index: r_k. The input $\mathbf{K}$ (knowledge) to our method is a set of prior distributions for some variables $r_{i,j}$. An example is shown in Table 1a(Top) expressing the belief that most likely there is a directed path from X to Y and from Y to Z.

The possible paths between variables dictate their possible **causal** and **associative** relations. When the Bayesian Network is interpreted causally, then $X \Rightarrow Y$ is equivalent to [X causes Y]. In addition, as discussed in the previous section: $X \Rightarrow Y$ or $X \Leftarrow Y$ or $X \Leftrightarrow Y$ is equivalent to [X is associated with Y]. Thus, a distribution $P_{r_{X,Y}} = \langle \pi_{\Rightarrow}, \pi_{\Leftarrow}, \pi_{\Leftrightarrow}, \pi_{\nLeftrightarrow}\rangle$ corresponds to the following beliefs about the causal and associative relations:

$$P(X \text{ causes } Y) = \pi_{\Rightarrow} \qquad P(X \text{ does not cause } Y) = \pi_{\Leftarrow} + \pi_{\Leftrightarrow} + \pi_{\nLeftrightarrow}$$
$$P(X \text{ associated with } Y) = \pi_{\Rightarrow} + \pi_{\Leftarrow} + \pi_{\Leftrightarrow} \qquad P(X \text{ not associated with } Y) = \pi_{\nLeftrightarrow}$$

In practice, it is useful to allow the user to specify prior beliefs directly on the events [X (not) causes Y] and [X is (not) associated with Y] from which the distribution $P_{r_{XY}}$ can be derived, than the opposite. This is not difficult: for example given $P(X \text{ causes } Y) = \pi_{\Rightarrow}$ the mass of probability $1 - \pi_{\Rightarrow}$ has to be distributed in a reasonable way to the other three values $\pi_{\Leftarrow}, \pi_{\Leftrightarrow}, \pi_{\nLeftrightarrow}$. However, we avoid this belief representation to simplify the presentation of the method.

Table 1: (a) (Top Part) Prior beliefs **K** regarding the paths between three pairs of variables. The beliefs are *incoherent*: $P(X \Rightarrow Y) = 0.8$ and $P(Y \Rightarrow Z) = 0.9$ imply that $P(X \Rightarrow Z) \in [0.7, 1]$. (a) (Bottom Part) Induced *coherent* beliefs $\mathbf{K}'$ stemming from **K** by solving the quadratic program in Eq. 10. (b) A part of the joint probability distribution J computed by solving Eq. 10 with input **K**. The number of DAGs with 5 nodes for each configurations N_C is also shown. The total number of DAGs with 5 vertices is $N = 29281$. The total number of configurations is $4^3 = 64$. Notice that C_2 and C_3 have both zero counts and zero probability, because they are invalid.

(a)

K	$\Rightarrow$	$\Leftarrow$	$\Leftrightarrow$	$\not\Leftrightarrow$
$r_{X,Y}(r_1)$	0.8	0.132	0.028	0.04
$r_{Y,Z}(r_2)$	0.9	0.066	0.014	0.02
$r_{X,Z}(r_3)$	0.6	0.264	0.056	0.08

$\mathbf{K}'$	$\Rightarrow$	$\Leftarrow$	$\Leftrightarrow$	$\not\Leftrightarrow$
$r_{X,Y}(r_1)$	0.705	0.175	0.051	0.069
$r_{Y,Z}(r_2)$	0.802	0.114	0.042	0.042
$r_{X,Z}(r_3)$	0.611	0.245	0.061	0.083

(b)

	$r_{X,Y}$	$r_{Y,Z}$	$r_{X,Z}$	p_C	N_C
C_1	$\Rightarrow$	$\Rightarrow$	$\Rightarrow$	0.5068	2800
C_2	$\Rightarrow$	$\Rightarrow$	$\Leftarrow$	0	0
C_3	$\Rightarrow$	$\Rightarrow$	$\Leftrightarrow$	0	0
...	...	...	...	...	...
C_{49}	$\not\Leftrightarrow$	$\Rightarrow$	$\Rightarrow$	0.0244	1045
...	...	...	...	...	...
C_{64}	$\not\Leftrightarrow$	$\not\Leftrightarrow$	$\not\Leftrightarrow$	$1.57 \cdot 10^{-9}$	309

4 Computing Priors and Scores

In this section, we derive a score $Sc(G|D, \mathbf{K})$ for a network graph G given data D and n prior distributions on paths beliefs in **K**. An important requirement for the computation of the score is knowledge of a joint distribution $J = P(r_1, \ldots, r_n) = P(\mathbf{r})$ such that its marginals correspond to the distributions in **K**. J assigns a probability value to each of the 4^n possible joint instantiations of values to variables $\mathbf{r} = \langle r_1, \ldots r_n \rangle$. We denote with C (configuration) such a given joint instantiation and define

$$p_C = P(\mathbf{r} = C|J)$$

In this section, we assume J is already computed; the next section describes the details of this computation. The joint J stemming from **K** in Table 1a(Top) is shown in Table 1b. It is important to notice that *for each graph G the configuration C is uniquely determined*. For example, in the joint of Table 1b, if in a graph G it holds $X \Rightarrow Y$, $Y \Rightarrow Z$, $X \Rightarrow Z$ then $\mathbf{r} = C_1$. Thus, it makes sense to denote with C_G the joint instantiation of variables **r** in graph G.

Let G be a Bayesian Network graph and D a dataset over the same variables. We now compute the probability $P(G|D, J)$:

$$P(G|D, J) = \frac{P(D|G, J) \cdot P(G|J)}{P(D|J)} = \frac{P(D|G) \cdot P(G|J)}{P(D|J)}$$

The second equation stems from the fact that given the graph G the data D are independent of J (J does not provide any additional information about the data once the graph is known). The factor $P(D|J)$ is a normalizing constant that does not need be computed when we maximize the above equation over different graphs. The factor $P(D|G)$ is the likelihood of the data given the graph; in Section 2 we mention several approximations (e.g., BDeu) based on different set of assumptions for each computation. We now focus on the prior $P(G|J)$:

$$P(G|J) = \sum_C P(G, C|J) = P(G, C_G|J)$$

The last equation holds because $P(G, C|J)$ equals zero for all $C \neq C_G$, since each graph entails exactly one configuration. Subsequently:

$$P(G|J) = P(G, C_G|J) = P(G|J, C_G) \cdot P(C_G|J) = P(G|C_G) \cdot P(C_G|J) = P(G|C_G) \cdot p_{C_G}$$

The factor $P(G|C_G)$ is our prior on a graph G given that a specific configuration holds. Given no other preference or knowledge *we assign the same (uniform) prior to all graphs with the same configuration*. Thus, letting N_C be the number of graphs over nodes $\mathcal{V}$ sharing the same configuration C then $P(G|C_G) = 1/N_{C_G}$ and so :

$$P(G|J) = \frac{p_{C_G}}{N_{C_G}} \quad \text{and} \quad Sc(G|J) = \log p_{C_G} - \log N_{C_G} \tag{2}$$

Similarly to Eq. 1 the overall score of a graph is:

$$Sc(G|D, J) = Sc(D|G) + Sc(G|J) \tag{3}$$

The score $Sc(G|D, J)$ has two desirable properties:

1. **Markov-Equivalent graphs that satisfy the same path-beliefs obtain the same score.** The last term in the equation above is the same for graphs sharing the same configuration. The first term is the same for Markov-equivalent graphs provided one employs an appropriate scoring function, such as the BDe and BDeu scores [8].
2. **For uninformative prior beliefs, all graphs are equiprobable**, i.e., $P(G|J) = 1/N$, where N is the number of graphs over nodes $\mathcal{V}$. With uninformative beliefs we expect to encounter a given configuration with probability equal to the proportion of the graphs satisfying the configuration, i.e,. $p_C = \frac{N_C}{N}$. In that case, $P(G|J) = \frac{N_C}{N} \cdot N_C = \frac{1}{N}$ and we end up with uniform priors as we would expect.

While Eq. 2 follows the above two properties, we point out to the fact that the factor $1/N_{C_G}$ may seem to provide counter-intuitive results at a first glance. Let's assume that for configurations C_1, C_2, the following holds: $p_1 = 0.6$ and $p_2 = 0.2$. In other words, the prior beliefs state that it is 3 times more probable a priori that the true graph has configuration C_1 than C_2. Now, let us assume that $N_1 = 60$ and $N_2 = 10$ and let G_1, G_2 be two graphs consistent with configurations C_1, C_2 respectively. We then obtain:

$$\frac{P(G_1|J)}{P(G_2|J)} = \frac{p_1 \cdot N_2}{p_2 \cdot N_1} = \frac{0.6 \times 10}{0.2 \times 60} = \frac{1}{2}$$

Thus, any graph consistent with C_2 has twice the prior than any graph in C_1. This may seem counter-intuitive since the user has specified that C_1 is 3 times more likely to be encountered than C_2. This is true considering the total probability mass of C_1 and C_2. However, since this mass is distributed over more graphs consistent with C_1 than C_2, each individual graph in the first configuration is less probably than any graph in the second configuration.

The implications of the above observation is that, everything else being equal, higher priors will tend to be assigned to graphs in "small" configurations, i.e., consistent with only a few graphs . If this behavior is not desirable then one can drop the $1/N_C$ factor and use:

$$P(G|J) = p_{C_G} \quad \text{and} \quad Sc(G|J) = \log p_{C_G} \tag{4}$$

However, if this score is used in place of Eq. 2 then Property 2 above is not satisfied any more.

Computing the number of graphs N_C. The number N of DAGs over nodes $\mathcal{V}$ has been solved in closed-form [14]. However, there is no closed-form to the best of our knowledge for the number N_C of DAGs that satisfy certain path-constraints. When the number of nodes is small (up to 5-6) one can enumerate all DAGs and compute each N_C for each configuration C by counting. The number of possible DAGs however, grows super-exponentially to the number of nodes and complete enumeration is not an option. In this case, we estimate these counts by sampling a number S of random DAGs with uniform probability. Specifically, we implemented the recent method in [15] that unlike prior work [16], avoids the use of expensive Markov-Chain, Monte-Carlo methods to ensure uniform sampling from the space of DAGs. $\hat{N}_C$ can be estimated as $\frac{S_C}{S}N$, where S_C is the number of sampled DAGs that conform to configuration C. When the number of configurations is large or N_C/N is small one may never sample any graph consistent with C. To avoid zero estimates, we apply the Laplace correction: $\hat{N}_C = \frac{S_C+l}{S+cl}N$, where c is the number of configurations and l an arbitrary parameter (we use the value $l = 1$).

5 Computing the Joint Distribution J given Prior Path Beliefs K

Eq. 2 shows how to compute the prior probability of a graph given the joint distribution J of path beliefs $\mathbf{r}$. In this section, we show how to compute J given the marginal beliefs on paths involving pairs of variables stored in $\mathbf{K}$. We denote with $\pi_{k,j}$ the probability that r_k takes value $j \in \{\Rightarrow, \Leftarrow, \Leftrightarrow, \nLeftrightarrow\}$:

$$\pi_{k,j} = P(r_k = j)$$

(a) (b)

Figure 1: We assume the prior beliefs **K** in Table 1a(Top) and the corresponding J in Table 1b. (a) The configuration $C_1 = \{X \Rightarrow Y, Y \Rightarrow Z, X \Rightarrow Z\}$ holds in the graph. For $p_1 = 0.5068$ (see Table 1b) we obtain the score $Sc(G|\mathbf{K}) = \log(0.5068) - \log(2800) = -8.6171$. (b) The configuration $C_{49} = \{X \nRightarrow Y, Y \Rightarrow Z, X \Rightarrow Z\}$ holds in the graph. For $p_{49} = 0.0244$ we obtain the score $Sc(G|\mathbf{K}) = \log(0.0244) - \log(1045) = -10.6662$. As expected, the first graph has a higher prior than the second one since $X \Rightarrow Y$ is given a higher probability than $X \nRightarrow Y$ in Table 1a(Top).

The values π are provided in **K**. The *unknown quantities* are p_C for each configuration C in J. Let $\mathcal{C}_{k,j} = \{C, \text{ s.t. } r_k = j\}$, i.e., the set of configurations where variable r_k obtains value j. For each k and j we obtain the following constraints:

$$\pi_{k,j} = \sum_{C \in \mathcal{C}_{k,j}} p_C \tag{5}$$

In other words, the marginals of the joint should equal our input path beliefs. An important observation that is characteristic of this problem, is that *path beliefs are not independent in general.* For example if one believes with certainty $X \Rightarrow Y \Rightarrow Z$, then they have to believe $X \Rightarrow Z$ to be coherent. Thus, it is important to consider the following constraints, stemming from the path semantics of the variables **r**:

$$p_C = 0, \text{ when } C \text{ is invalid} \tag{6}$$

By invalid we mean a configuration that cannot be satisfied by the graph of any Bayesian Network over $\mathcal{V}$, e.g., it contains directed cycles. The algorithm to detect invalid configurations is discussed later. To complete the problem specification we impose that:

$$\sum_C p_C = 1 \quad \text{and} \quad p_C \geq 0 \tag{7}$$

If constraints in Eqs. 5, 6, 7 can be satisfied then a joint distribution adhering to the probability axioms can be found such that the prior marginal path beliefs hold. In this case, by definition **K** is *coherent*, otherwise it is *incoherent*. Notice that all constraints together form a set of linear equations that is easy to solve or determine it has no (non-negative) solution. However, the number of unknowns p_C equals 4^n, where n are the input path beliefs and so the computational overhead increases exponential with n.

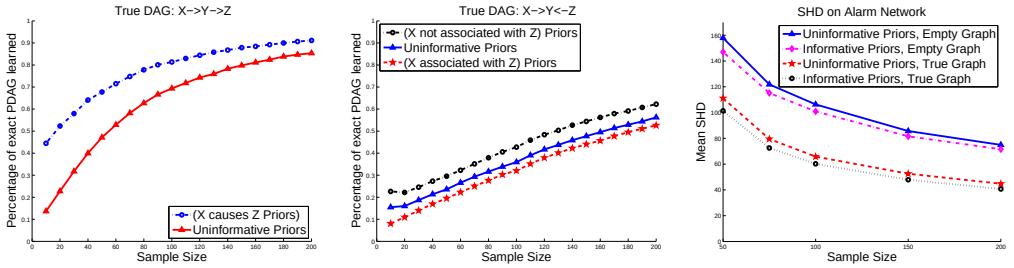
Dealing with Coherent Beliefs. The systems of equations contains $4n$ constraints from Eq. 5, m constraints from Eq. 6 and 1 constraint from Eq. 7 and 4^n unknowns. For most typical problems, $4n + m + 1 \ll 4^n$ and so the system may have infinite solutions. We argue that one should choose a solution jpd J as close to the uninformative one as possible. Any other distribution may introduce bias towards certain configurations, even if the prior knowledge does not suggest preference over those configurations. In other words, if the uninformative jpd is a coherent extension of the prior knowledge, there is no reason to prefer any other solution over it. The problem can be formulated as follows:

$$\min_{\mathbf{P}} \sum_{k=1}^{4^n} (p_k - \frac{N_k}{N})^2 \text{ subject to constraints in Eqs. 5, 6, 7} \tag{8}$$

The quantity $\frac{N_k}{N}$, where N_k is the number of graphs consistent with configuration C_k and N the total number of DAGs over $\mathcal{V}$ corresponds to the uninformative priors where each graph is equiprobable. The optimization problem of Eq. 8 is a quadratic program (quadratic objective function with linear constraints) and can be solved accurately and relatively efficiently (to the number of unknowns).

Dealing with Incoherent Beliefs. In this case, there is no jpd that can equal the marginal input beliefs. Instead of requesting coherent beliefs or ignoring the incoherency, we seek for joints with marginals as close as possible to the user's input beliefs. The constraints in Eq. 5 are now modified to include slack variables $s_{i,j}$, i.e., the amount by which the original constraints are violated:

$$\pi_{k,j} + s_{k,j} = \sum_{C \in \mathcal{C}_{k,j}} p_C \tag{9}$$



(a) (b) (c)

Figure 2: Proof-of-concept, experimental results. (a) Learning the orientations and the skeleton is facilitated by causal prior knowledge. (b) Learning the graph is facilitated by correct associative prior knowledge and hindered by incorrect priors. (c) Learning the ALARM network with 5 pieces of informative associative beliefs and without.

This system of equations is always solvable; out of all solutions preference should be given to solutions that violate the original constraints the least, leading to the following optimization problem:

$$\min_{\mathbf{p},\mathbf{s}} \sum_{l=1}^{4n} s_l^2 + \alpha \cdot \sum_{k=1}^{4^n} (p_k - \frac{N_k}{N})^2 \text{ subject to constraints in Eqs. 9, 6, 7} \tag{10}$$

This problem tries to simultaneously minimize a trade-off between (a) the difference between the marginal probabilities and the user beliefs and, (b) the difference between the solution jpd and the uninformative jpd. The trade-off is controlled by the parameter α. For $\alpha = 0$ one finds a valid jpd so that its marginals are as close as possible to the input beliefs. For $\alpha = 4n/4^n$ (the ratio of terms in each summand) each summand is assigned equal importance (this is the value we employ in our experiments). Table 1b contains the joint J stemming from $\mathbf{K}$ of Table 1a(Top) computed by solving Eq. 10. For comparison with the input beliefs $\mathbf{K}$, Table 1a(Bottom) contains the marginal beliefs $\mathbf{K}'$ implied by J: $\pi'_{i,j} = \pi_{i,j} + s_{i,j}$. The values in Table 1a(Top) and Table 1a(Bottom) are close, with the later one representing coherent beliefs.

Determining Invalid Configurations. To identify all constraints in Eq. 6 we have implemented the following algorithm. For each configuration C, we construct a graph G' with nodes the variables that appear in at least one prior path belief. For each assignment $r_{XY} =$ " $\Rightarrow$ " or $r_{XY} =$ " $\Leftarrow$ " in C, we add the edge $X \rightarrow Y$ or $X \leftarrow Y$ respectively, in G'. In addition, for each assignment $r_{XY} =$ " $\Leftrightarrow$ " we add a new dummy node V_d to G' and add the edges $X \leftarrow V_d \rightarrow Y$. Configuration C is invalid in jthe following cases: (a) G' contains cycles, (b) for some X, Y in G', X has a directed path to Y and $r_{XY} =$ " $\not\Leftrightarrow$ " or $r_{XY} =$ " $\Leftrightarrow$ " in C, and (c) for some X, Y in G', X has a path to Y in G' (not necessarily directed) and $r_{XY} =$ " $\not\Leftrightarrow$ " in C.

This algorithm is obviously sound, but it is not complete. A problem may arise when the number of dummy nodes added to G' exceeds the number of available nodes (variables) in the data. In that case, it may seem that a configuration is valid, but there may not be enough variables to satisfy all confounding $\Leftrightarrow$ relations in the context of the remaining path constraints. The simplest example is a dataset with two variables X and Y: the configuration $r_{XY} =$ " $\Leftrightarrow$ " is invalid as there is no other variable to serve as common ancestor. Yet, the above cases will not identify it as such. A less trivial example is $r_{XY} =$ " $\Leftrightarrow$ " and $r_{YZ} =$ " $\Rightarrow$ " when the only variables are X, Y, Z. Since $r_{XY} =$ " $\Leftrightarrow$ " it has to be that $X \Leftarrow Z \Rightarrow Y$ which conflicts with $r_{YZ} =$ " $\Rightarrow$ ". Our intuition is that a complete algorithm requires solving a constraint satisfaction problem. However, when the number of variables in the data is large relative to the number of path beliefs (specifically if $|V_{data}| \geq |V_{G'}|$ holds), the algorithm becomes complete (proof omitted for space).

6 Experimental Results

Employing Causal Knowledge. We consider the graph $X \rightarrow Y \rightarrow Z$. As prior knowledge we set $P(X \Rightarrow Z) = 0.9$ and distribute the remaining 0.1 mass of probability to the remaining values of r_{XZ} proportional to the values that correspond to a uniform prior. We repeat the following

experiment 10000 times: (a) we randomly select the number of states for each variable to be either 3 or 4, (b) we sample the cpts for each variable using the gamma distribution ($gamrnd$ Matlab function with shape parameter A set to 0.5 and scale parameter B set to 1), (c) we sample a dataset of size 200 from the network given the previously sampled cpts, (d) we increase the samples of the dataset to provide to the scoring method from 10 to 200 with step size of 10, (e) we identify the highest scoring network out of all 25 possible DAGs using informative priors and the BDeu score with Equivalent Sample Size (ESS) set to 1 (see Eq. 3), (f) we similarly identify the highest scoring network with uniform priors.

Results: Figure 2a plots the percentage of the time the PDAG $X - Y - Z$ of the true network was found exactly with and without informative priors. First notice, that when the true PDAG is found exactly, the edges are also *always oriented correctly since the true network has a higher prior than any other Markov-equivalent graph*. Perhaps more surprising though, notice that the informative priors also *increase the learning of the skeleton.* The belief $X \Rightarrow Z$ tends to add a path from X to Z. The associations $X - Y$ and $Y - Z$ are always higher than or equal to the association between $X - Z$ (see [17]). Thus, *it is the correct path $X - Y - Z$ that tends to be induced, rather than any other network with a path $X \Rightarrow Z$.*

Employing Associative Knowledge. We run a similar proof-of-concept experiment where the true network is a single collider $X \rightarrow Y \leftarrow Z$. We use the same settings as before for three cases: correct associative priors $P(X \nLeftrightarrow Z) = 0.9$, uniform priors, and incorrect associative priors $P(X$ associated with $Z) = 0.9$.

Results: The results are shown in Figure 2b. As expected, *correct prior beliefs clearly improve the chances of identifying the true PDAG; the effect is exactly the opposite when misleading, incorrect beliefs are provided to the algorithm*. Of course, asymptotically the priors, whether correct, incorrect, or uninformative play no role.

Learning Larger Networks. We sample 1000 datasets from the distribution of the ALARM network [18]. We learnt the network using greedy search-and-score with the typical operators add, delete, and reverse an edge, and the BDeu metric with ESS=1. We vary the sample size given to the algorithms within $\{50, 75, 100, 150, 200\}$. For each dataset, we randomly pick 5 pairs (X, Y) of variables on which to provide informative associative priors: if $X \nLeftrightarrow Y$ in the true network, we set $P(X \nLeftrightarrow Y) = 0.9$, otherwise, we set $P(X \nLeftrightarrow Y) = 0.1$. We run search-and-score starting from the empty graph with and without the informative priors and compute the Structural Hamming Distance [19] from the true network. The simple search operators do not consider and neither exploit the path beliefs to improve optimization. We thus, also run the search-and-score algorithm starting from the true network to gauge the potential for improvement when a better search method is employed, that at some point visits the true network.

Results: The results are shown in Figure 2c. *In both cases, the SHD is smaller with the informative priors than with uniform priors.* The differences in SHD for each sample size are always statistically significant (using a one-sample t-test), with p-value close to the machine epsilon. For low sample sizes (50 and 75) the 95% confidence interval of the SHD differences are $[10.0959, 11.7821], [6.1051, 7.2349]$ when starting from the empty graph, and $[8.8170, 10.6630], [6.2721, 7.5399]$ when starting from the true graph.

7 Discussion and Conclusions

We present a method for computing informative priors given a set of causal and associative beliefs on pairs of variables. The priors can then be employed by any search-and-score learning algorithm. Such beliefs can be induced from prior experimental or observational studies respectively, among other sources. The method, for the first time, addresses the issues of incoherent priors and priors that are not independent. Providing correct priors about pairwise causal or associative relations improves learning both in terms of identifying the orientation of the edges (for causal priors), but also in terms of identifying the skeleton of the network.

There are numerous issues to still address regarding both the method and the general problem. The algorithm computes a joint of prior beliefs that is exponential to the input (number of beliefs). More efficient algorithms that perform this operation implicitly are desirable. The search method for the optimal graph, in the context of informative priors becomes more complicated; typical greedy-

search with operators on the edges alone may not suffice. Complete and efficient algorithms for determining invalid configurations, as well as closed-form solutions for computing the number of graphs given path constraints are desirable. Finally, incorporating the strength of the causal effects or associations and other prior knowledge characteristics is an interesting future direction to pursue.

References

[1] G. F. Cooper and E. Herskovits. A bayesian method for the induction of probabilistic networks from data. *Mach. Learn.*, 9:309–347, 1992.

[2] P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. MIT Press, Cambridge, MA, 2nd edition, 2000.

[3] P. Hansen, B. Jaumard, M. Poggi de Aragão, F. Chauny, and S. Perron. Probabilistic satisfiability with imprecise probabilities. *International Journal of Approximate Reasoning*, 24(2-3):171–189, 2000.

[4] R. S. Niculescu, T. M. Mitchell, and R. B. Rao. Bayesian network learning with parameter constraints. *JMLR*, 7:1357–1383, 2006.

[5] C. Meek. Causal inference and causal explanation with background knowledge. In *UAI*, pages 403–418. Morgan Kaufmann, August 1995.

[6] W. Buntine. Theory refinement on bayesian networks. In *Proceedings of the seventh conference (1991) on Uncertainty in artificial intelligence*, pages 52–60, San Francisco, CA, USA, 1991. Morgan Kaufmann Publishers Inc.

[7] C. Robert and S. Arno. Priors on network structures. biasing the search for bayesian networks. *Int. J. Approx. Reasoning*, pages 39–57, 2000.

[8] D. Heckerman, D. Geiger, and D. M. Chickering. Learning bayesian networks: The combination of knowledge and statistical data. *Mach. Learn.*, 20(3):197–243, 1995.

[9] P. Spirtes. Introduction to causal inference. *JMLR*, 11:1643–1662, 2010.

[10] G. Borboudakis, S. Triantafillou, V. Lagani, and I. Tsamardinos. A constraint-based approach to incorporate prior knowledge in causal models. *ESANN*, 2011.

[11] R. T. O'donnell, Ann E. Nicholson, B. Han, Kevin B. Korb, M. J. Alam, and L. R. Hope. Incorporating expert elicited structural information in the CaMML causal discovery program. *AI'06 Proceedings of the 19th Australian joint conference on Artificial Intelligence: advances in Artificial Intelligence*, pages 1–16, 2006.

[12] J. Pearl. *Causality, Models, Reasoning, and Inference*. Cambridge University Press, Cambridge, U.K., 2000.

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

[14] R. W. Robinson. Counting Labeled Acyclic Digraphs. In F. Harary, editor, *New Directions in Graph Theory*. New York: Academic Press, 1973.

[15] J. Kuipers and G. Moffa. Uniform generation of random acyclic digraphs. *ArXiv e-prints*, February 2012.

[16] G. Melancon, M. Bousquet-Melou, and I. Dutour. Random generation of dags for graph drawing. Technical report, CWI, Stichting Mathematisch Centrum, 2000.

[17] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, 2006.

[18] I. Beinlich, G. Suermondt, R. Chavez, and G. Cooper. The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. In *2nd European Conference in Artificial Intelligence in Medicine*, pages 247–256, 1989.

[19] I. Tsamardinos, L.E. Brown, and C.F. Aliferis. The Max-Min Hill-Climbing Bayesian Network Structure Learning Algorithm. *Machine Learning*, 65(1):31–78, 2006.