Discovery and density estimation of latent confounders in Bayesian networks with evidence lower bound

Kiattikun Chobtham k.chobtham@qmul.ac.uk

Anthony C. Constantinou a.constantinou@qmul.ac.uk

Bayesian Artificial Intelligence research lab, School of Electronic Engineering and Computer Science, Queen Mary University of London, London, UK, E1 4NS.

Abstract

Discovering and parameterising latent confounders represent important and challenging problems in causal structure learning and density estimation respectively. In this paper, we focus on both discovering and learning the distribution of latent confounders. This task requires solutions that come from different areas of statistics and machine learning. We combine elements of variational Bayesian methods, expectation-maximisation, hill-climbing search, and structure learning under the assumption of causal insufficiency. We propose two learning strategies; one that maximises model selection accuracy, and another that improves computational efficiency in exchange for minor reductions in accuracy. The former strategy is suitable for small networks and the latter for moderate size networks. Both learning strategies perform well relative to existing solutions.

Keywords: Ancestral graphs; EM algorithm; greedy search; hill-climbing search; hidden variables; probabilistic graphical models; variational inference.

1. Introduction

Methods from both machine learning and statistical sciences have contributed to the advancements in probabilistic graphical models, and specifically in learning Bayesian Networks (BNs). The structure of a BN is typically represented by a Direct Acyclic Graph (DAG), containing nodes that represent variables and edges that represent conditional relationships. The process of learning BNs from data is separated into two tasks. An unsupervised structure learning approach is first used to determine the graph of the BN, followed by parameter estimation of the conditional distributions given the learnt structure.

The task of structure learning is known to be both difficult and computationally expensive, and it is NP-hard even for small networks containing just 10 variables. These challenges are elevated when noise is present in the data, or when the data are incomplete (Constantinou et al., 2021). One such example is when the input data do not capture all the variables; often referred to as the problem of learning under the assumption of causal insufficiency (Spirtes et al., 2001). A special case of a latent variable is the latent confounder which represents a missing common cause of two or more observed variables, and tends to lead to spurious edges between its children.

Structure learning algorithms that assume causal insufficiency generate ancestral graphs that represent an extension of a DAG. These algorithms generate either a Maximal Ancestral Graph (MAG) that contains bi-directed edges indicating confounding and directed edges indicating causal or ancestral relationships, or a Partial Ancestral Graph (PAG) that represents the Markov equivalence class of a set of MAGs, in the same way a Completed

Partial DAG (CPDAG) represents the Markov equivalence class of a set of DAGs. As we later explain in subsection 2.1, a PAG also captures the possible existence of multiple latent confounders. Well-established algorithms that generate PAGs tend to be constraint-based or hybrid, and largely rely on FCI by Spirtes et al. (2001). Some variants of FCI include the constraint-based RFCI algorithm by Colombo et al. (2012), cFCI by Ramsey et al. (2012), mFCI by Colombo and Maathuis (2014), and the hybrid GFCI algorithm by Ogarrio et al. (2016) which combines elements of constraint-based and score-based learning.

In this work, we describe two learning strategies that take a PAG as an input, along with observed data, to determine the existence of latent confounders and learn their underlying latent distributions. We propose two algorithms: one that maximises accuracy and another that balances accuracy with computational complexity. The paper is organised as follows: Section 2 provides preliminary information through past related works, Section 3 describes the two algorithms, Section 4 describes the experimental setup, Section 5 presents the results, and we provide our conclusions and directions for future work in Section 6.

2. Preliminaries

2.1 Ancestral Graphs

As discussed in the introduction, a MAG represents a DAG extension that captures information about possible latent confounders. A PAG, which represents a set of Markov equivalent MAGs that entail the same set of Conditional Independencies (CIs) or m-separation criteria, is represented by a tuple (V, E) where V is the set of observed variables and E is the set of the edges. The edges in a PAG can be: —, $\leftrightarrow$, $\rightarrow$, o$\rightarrow$ or o—o, where — indicates selection variables, $\leftrightarrow$ indicates latent confounders, and $\rightarrow$ indicates that all MAGs in the equivalence class contain this directed edge. The circle edge mark (o) indicates that the endpoint of the edge could be either a tail (–) or an arrowhead (>). For example, o$\rightarrow$ indicates that the corresponding MAGs will contain either $\rightarrow$ or $\leftrightarrow$, whereas o—o indicates that the edge can be $\rightarrow$, $\leftarrow$ or $\leftrightarrow$. Because we do not deal with selection bias in this paper, we will not be considering ancestral graphs that contain undirected edges (—). Both PAGs and MAGs are acyclic and do not assume partially directed cycles when $A \leftrightarrow B$; but instead assume that either B is an ancestor of A, or A is an ancestor of B (Richardson and Spirtes, 2000).

2.2 Conjugate-exponential family models

We consider conjugate-exponential family models for discrete data. We assume a Dirichlet prior that serves as a conjugate prior of a multinomial likelihood (Bishop, 2006), whose posterior distribution is also Dirichlet. We use the empirical Bayes method by Gelman et al. (2003) to determine the prior parameters from data. For density estimation of latent confounders, we assume a Dirichlet prior $Dir\left(\theta_i|\alpha_{ik}\right)$ where α_{ik} is a hyperparameter set to '1' for uniform distribution, and θ_i denotes parameters $\sum_k \theta_{ik} = 1$ where k represents the number of states. Since we perform structure learning and density estimation under causal insufficiency, some variables will not be observed in the data, leading to an incomplete-data marginal likelihood $p(D|G)$ of a DAG G.

2.3 Variational Bayesian Expectation-Maximization (VBEM) algorithm

Marginalising out the parameters over latent confounders L_j in $p(D|G)$ makes the task of learning prohibitively expensive and intractable. We address this issue by approximating distributions of latent variables using the computationally efficient Variational Bayesian Expectation-Maximization (VBEM) algorithm (Beal and Ghahramani, 2003) that enables tractable solutions. The VBEM algorithm combines elements of variational inference (Jordan et al., 1999) and Expectation-Maximisation (EM; Friedman, 2013). It uses an alternated optimisation technique to find a surrogate distribution $q(L, \theta)$ from any exponential family Q (e.g., Gaussian, Dirichlet, multinomial) and optimises towards the true distribution $p(L, \theta|D, G)$. VBEM offers an approximate solution that guarantees to monotonically increase the objective score, and scales better with large data compared to MCMC (Hastings, 1970).

The objective of VBEM is to minimise the discrepancy between two distributions $q(L, \theta)$ and $p(L, \theta|D, G)$. It uses the reverse Kullback-Leiber (KL) divergence for this task, which is the standard choice for variational inference, defined as follows:

$$\begin{aligned} KL(q \parallel p) &= \iint dLd\theta q(L, \theta) log\frac{q(L,\theta)}{p(L,\theta|D,G)} \\ &= \mathbb{E}_q\left[log\frac{q(L,\theta)}{p(L,\theta|D,G)}\right] \\ &= \mathbb{E}_q[logp(D|G)] - \{\mathbb{E}_q[logp(L, \theta, D|G)] - \mathbb{E}_q[logq(L, \theta)]\} \quad (1) \end{aligned}$$

Because the incomplete-data marginal likelihood $p(D|G)$ is intractable to compute, we consider $p(D|G)$ to be a constant. The aim is to minimise $KL(q \parallel p)$, which is equivalent to maximising the Evidence Lower Bound (ELBO). Therefore, we can minimise $KL(q \parallel p)$ without having to know the true distribution $p(L, \theta|D, G)$ and $p(D|G)$. We can describe ELBO as the objective function:

$$\text{ELBO} = \mathbb{E}_q[logp(L, \theta, D|G)] - \mathbb{E}_q[logq(L, \theta)] \quad (2)$$

where $q(L, \theta)$ is assumed to be the factorisation of the free distributions $q_L(L)$ and $q_\theta(\theta)$. We maximise ELBO using a function $\mathcal{F}$ of both $q_L(L)$ and $q_\theta(\theta)$ as follows (Beal and Ghahramani, 2006):

$$\text{ELBO} = \mathcal{F}(q_L(L), q_\theta(\theta)) = \iint dLd\theta q_L(L) q_\theta(\theta)[logp(L, \theta, D|G) - log(q_L(L) q_\theta(\theta))] \quad (3)$$

To maximise $\mathcal{F}$, VBEM calculates $q_L(L)$ and $q_\theta(\theta)$ while holding the other fixed at iteration t. The two steps for each iteration t are:

1) VB-E step: estimates the posterior distribution over latent confounders $q_L^{t+1}(L) = \prod_{i=1}^{|L|} q_{L_i}^{t+1}(L_i)$ given $q_\theta^t(\theta)$ from the last iteration by taking the functional derivatives in Equation (3) with respect to $q_{L_i}(L_i)$, where $|L|$ is the number of latent confounders.

2) VB-M step: estimates $q_\theta^{t+1}(\theta)$ given the posterior distribution ${q_L}^{t+1}(L)$ taken from the VB-E step by taking the functional derivatives in Equation (3) with respect to $q_\theta(\theta)$.

VBEM iterates over the VB-E and VB-M steps until the difference in ELBO becomes smaller than a given threshold, indicating convergence. Since ELBO is not a score-equivalent function, it generates different values for graphs that belong to the same Markov equivalence class. A revised version called p-ELBO was proposed by Rodriguez-Sanchez et al. (2020) that includes a penalty term to avoid the $|L_i|!$ equivalent ways of assigning sets of parameters that result in the same distribution (non-identifiability), and it is defined as p-ELBO = ELBO $- \sum_{i=1}^{|L|} log\ \ |L_i|!$; where $|L_i|$ is the number of states in L_i.

2.4 Past relevant work

ELBO was used as the objective function of a neural network in Variational Autoencoder (VAE) by Kingma and Welling (2013). VAE for heterogeneous Mixed type data (VAEM) was used by Ma et al. (2020) for density estimation of latent variables in deep generative models. VAE assumes each observed variable has a latent parent, whereas VAEM is an extension of VAE that assumes an additional latent confounder that serves as a parent of all latent variables.

The ELBO score was extended to p-ELBO by Rodriguez-Sanchez et al. (2020; 2022), which was used as the objective score in Constrained Incremental Learner (CIL) and Greedy Latent Structure Learner (GLSL) algorithms. CIL learns a tree-structured BN that assumes any two nodes are connected by one directed path only, whereas GLSL learns a DAG BN. Both algorithms start from an empty graph and perform various search operations including a) add or remove latent variables as parents of observed variables, b) increase the number of states of latent variables, and c) perform edge operations such as add, remove, or reverse edges, aiming to maximise p-ELBO. Searching for latent confounders often means iterating over all pairs of observed variables, which can be computationally expensive. Instead, these algorithms offer a strategy that focuses on a set of pairs of variables that provide the highest Mutual Information (MI). Empirical results show that GLSL outperforms CIL, but at the expense of high computational complexity.

3. Two new algorithms for learning latent confounders

This section describes the two learning strategies we have implemented for latent confounder discovery and density estimation. Subsection 3.1 describes how we use existing algorithms to draw a PAG that is then given as an input to the two algorithms we propose, which in turn use the PAG to search for different MAGs and DAGs with parameterised latent confounders. We describe the two algorithms in subsections 3.2 and 3.3 respectively. Both algorithms assume the input data are discrete, and that the latent confounders have no parents but have at least two children. We further assume a Dirichlet prior $q_\theta(\theta)$ over all parameters as described in subsection 2.2, and we use p-ELBO as the objective function which is computed using the VBEM algorithm as described in subsection 2.3.

3.1 Searching for MAGs and DAGs given a PAG input

The FCI algorithm and some of its variants discussed in the introduction represent the state-of-the-art in recovering ancestral graphs under the assumption of causal insufficiency (Kitson et al., 2021). Any of these algorithms can be used to draw a PAG that can be

given as input to the two algorithms described in subsections 3.2 and 3.3. A set of Markov equivalent MAGs can be then derived from the input PAG. However, because the number of possible latent confounders that can be explored for a given MAG is generally intractable, we shall assume the minimum number of latent confounders that satisfy the m-separation criteria. Since the computational cost of working with multiple latent confounders is high, it becomes necessary that we introduce Assumption 1, as described below.

Assumption 1: The optimal number of latent confounders is the minimum number of latent confounders that retain the CIs of a given MAG.

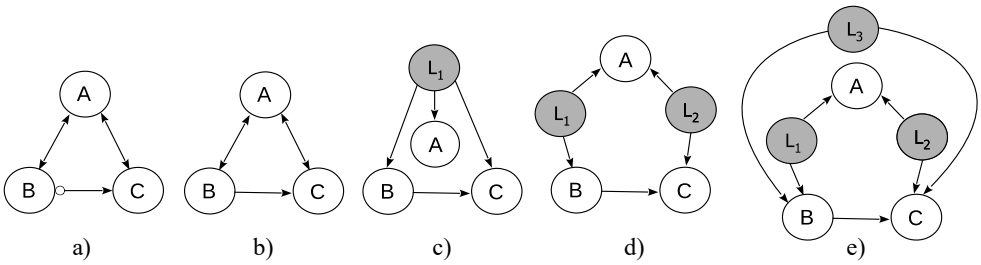

Figure 1. A PAG (a) along with one of its MAGs (b), and three DAGs (c, d, e) with different latent confounders (grey nodes) derived from the given MAG, where $A \not\perp B$, $A \not\perp C$ and $B \not\perp C$.

Figure 1 presents a simple PAG that contains two bi-directed edges, along with a MAG and three DAGs that satisfy the CI statements of the PAG. Converting a MAG into possible DAGs implies that each DAG retains the CIs of that MAG by reducing the criteria of m-separation to d-separation. In this example, the DAG that contains the minimum number of latent confounders, with reference to the MAG in Figure 1b, is shown in Figure 1c. The DAGs in Figures 1d and 1e contain a higher number of latent confounders than the minimum required to satisfy all the CIs of the given MAG. Because the algorithms we describe in subsection 3.2 and 3.3 rely on **Assumption 1**, they will never explore DAGs that contain a higher number of latent confounders than the minimum required, and would not visit DAGs such as those shown in Figures 1d and 1e.

3.2 Alg 1: Incremental Latent Confounder search with VBEM (ILC-V)

The first algorithm, which we call Incremental Latent Confounder search with VBEM (ILC-V), is described in Algorithm 1. It takes a PAG input (Step 1) and uses the ZML algorithm available in R (Malinsky and Spirtes, 2017) to enumerate all Markov equivalent MAGs of that PAG (Step 3). It then further constructs DAGs for each MAG, starting from the MAGs that contain the minimum number of bi-directed edges (Step 4). Each latent confounder modelled at Step 4 is assumed to be binary, and the optimal DAG is the one that maximises p-ELBO using the VBEM algorithm made available as a Java library by Rodriguez-Sanchez (2021).

At Step 5, Algorithm 1 calls Algorithm 1b to determine the optimal number of states for each latent confounder. This is achieved by iterating over each latent confounder present in the highest scoring DAG determined at Step 4, and greedily increasing the number of states by one at a time, for each latent confounder. Algorithm 1b returns a DAG that contains the optimal number of states for each latent confounder, or the maximum number of states S if the objective score continues to increase with the number of states. To

improve computational complexity, the objective score p-ELBO is applied to a subgraph G_S that contains the auxiliary latent confounders and their children, since the conditional distributions of the remaining nodes remain unchanged in the BN. The final Step 6 generates the final DAG BN and revises the p-ELBO score.

Algorithm 1: Incremental Latent Confounder search with VBEM (ILC-V)

Input: A structure learning algorithm that generates PAG, max Sepset size n, significant threshold α, observational data D, converge threshold c, max number of bi-directed edges m, a runtime limit T.
Output: A DAG BN that contains latent confounders as observed variables, along with the conditional distributions.

```
Step 1  PAG ← Run a structure learning algorithm with α and n given D
Step 2  S ← max number of states in D
        current number of bi-directed edges ← count the total number of bi-directed edges in PAG
        score_improve = TRUE
        best_pELBO = - Infinity
Step 3  List of MAGs ℒ_M ← Enumerate all Markov equivalent MAGs from PAG
Step 4  While score_improve = TRUE or current number of bi-directed edges ≤ m or elapsed time ≤ T
            best_local_pELBO = - Infinity
            For each MAG in ℒ_M where its #bi-directed edges = current number of bi-directed edges
                Construct new DAG G that contains all edges → present in MAG and generate boolean
                    auxiliary latent confounders for edges ↔ present in MAG as per Assumption 1
                current_pELBO ← run VBEM until p-ELBO converges with c given D and G
                If current_pELBO > best_pELBO
                    best_pELBO = current_pELBO
                    best_DAG = G
                If current_pELBO > best_local_pELBO
                    best_local_pELBO = current_pELBO
            current number of bi-directed edges++
            If best_pELBO > best_local_pELBO
                score_improve = FALSE
Step 5  If S > 2
            get best_DAG with (potentially) multinomial latent confounders ← run Algorithm 1b given best_DAG,
            D, t and S
Step 6  get best_pELBO and return Output ← run VBEM until p-ELBO converges with c given D and best_DAG
```

Algorithm 1b: Greedy search for the optimal number of states for each latent confounder

Input: A DAG G with auxiliary boolean latent confounders, max states S for each latent confounder, observational data D, converge threshold c.
Output: A DAG G with auxiliary (potentially) multinomial latent confounders.

```
Step 1  score_improve = TRUE
        best_pELBO = - Infinity
Step 2  For each latent confounder i in DAG G
            While score_improve = TRUE or number of states ≤ S
                current_pELBO ← run VBEM until p-ELBO converges with c given D and subgraph G_S
                If current_pELBO > best_pELBO
                    best_pELBO = current_pELBO
                Else
                    score_improve = FALSE
                    number of states of latent confounder i --
                number of states of latent confounder i ++
                Update the number of states of latent confounder i in G_S and G
Step 3  Return G with the optimal number of states for each latent confounder
```

3.3 Alg 2: Hill-Climbing Latent Confounder search with VBEM (HCLC-V)

Because ILC-V (Algorithm 1) is computationally expensive, as we later show in Section 5, one might be interested in a computationally efficient version that minimally decreases the

objective score of Algorithm 1. A problem with ILC-V is that when the input PAG contains a high number of invariant edges o—o or o→, enumerating all possible MAGs can quickly cause memory allocation problems. To address this, we introduce a modified version of ILC-V, which we call Hill-Climbing Latent Confounder search with VBEM (HCLC-V), that skips Markov equivalence checks. This means that HCLC-V no longer needs to check the CIs for each DAG visited, and this saves enormous computational time. Instead, HCLC-V iterates over possible edge orientations as described in Step 4 of Algorithm 2, and continues to follow the incremental search strategy of ILC-V in terms of the number of bi-directed edges. Moreover, a list of the best-found latent confounders from one iteration is carried forward to the next iteration (see Steps 3 and 4 in Algorithm 2). Lastly, since HCLC-V relies on hill-climbing search, it stops exploration when a local maximum is reached.

Algorithm 2: Hill-Climbing Latent Confounder search with VBEM (HCLC-V)

Input: A structure learning algorithm that generates PAG, max Sepset size n, significant threshold α, observational data D, converge threshold c, max number of bi-directed edges m, a runtime limit T.
Output: A DAG BN that contains latent confounders as observed variables, along with the conditional distributions.

```
Step 1  Same as in Algorithm 1
Step 2  Same as in Algorithm 1
Step 3  List of best_latent_confounder ℒ_L = ∅
Step 4  While score_improve = TRUE or current number of bi-directed edges ≤ m or elapsed time ≤ T
            best_local_pELBO = - Infinity
            While all pairs A o—o B in PAG are not orientated
                Construct new DAG G by changing all o→ present in PAG to → and generate boolean
                    auxiliary latent confounders for edges ↔ present in PAG as per Assumption 1
                Orientate A → B or A ← B in G from all pairs A o—o B with the maximum p-ELBO using VBEM
            For each pair A o—o B or A o→ B in PAG which is not in ℒ_L
                Construct new MAG that contains all edges → present in G and add the edge A ↔ B and others
                    C ↔ D given ℒ_L
                Construct new DAG G' that contains all edges → present in MAG and generate
                    boolean auxiliary latent confounders for edges ↔ present in MAG as per Assumption 1
                current_pELBO ← run VBEM until p-ELBO converges with c given D and G'
                If current_pELBO > best_pELBO
                    best_pELBO = current_pELBO
                    best_DAG = G'
                    Add the auxiliary latent confounders to ℒ_L
                If current_pELBO > best_local_pELBO
                    best_local_pELBO = current_pELBO
            current number of bi-directed edges++
            If best_pELBO > best_local_pELBO
                score_improve = FALSE
Step 5  Same as in Algorithm 1
Step 6  Same as in Algorithm 1
```

4. Case studies and evaluation setup

The experimental setup involves four real-world BNs taken from the Bayesys repository (Constantinou et al., 2020), described in Table 1. We generated synthetic data of 1k and 10k samples for each network using the bnlearn R package (Scutari, 2010). One data set is created for each latent confounder listed in Table 1. This process was applied to both sample sizes, and led to a total of 22 data sets.

We have used the constraint-based FCI and the hybrid GFCI algorithms to generate PAGs to be provided as input to ILC-V and HCLC-V. This produced four different result-combinations, which we refer to as ILC-V_{FCI}, HCLC-V_{FCI}, ILC-V_{GFCI} and HCLC-V_{GFCI}

in Section 5. The GFCI algorithm was tested using the Tetrad-based rcausal R package (Wongchokprasitti, 2019), and the FCI algorithm was tested using the pcalg R package (Kalisch et al., 2012). Regarding the hyperparameters of FCI and GFCI, we set the G-square significance threshold to α=0.05 and the Sepset size to n=-1 for unlimited size of conditioning sets. For ILC-V and HCLC-V, we set the maximum number of bi-directed edges to m=4 to enable us to carry out experiments within reasonable runtime, and the convergence threshold of VBEM to c=0.01.

BN	Variables	Edges	Max in-degree	Free parameters	Potential latent confounders
Asia	8	8	2	18	Smoke
Sports	9	15	2	1,049	RDlevel
Property	27	31	3	3,056	propertyPurchaseValue, borrowing, otherPropertyExpenses
Alarm	37	46	4	509	INTUBATION, HYPOVOLEMIA, LVFAILURE, ERRCAUTER, PULMEMBOLUS, KINKEDTUBE

Table 1. The properties of the four real-world networks considered for evaluation.

We assess the accuracy of ILC-V and HCLC-V in terms of the objective score p-ELBO and learning runtime, with reference to those obtained by the GLSL and CIL algorithms discussed in subsection 2.4. GLSL and CIL are tested using the Java library by Rodriguez-Sanchez (2021) with mi=10 regarding the number of pairs of variables to be considered with the highest MI, and with $maxNumberParents\_latent$=-1 for GLSL to assume no parents for density estimation of latent confounders to enable us to carry out experiments within reasonable runtime.

We impose a runtime limit of 12 hours for each experiment and set hyperparameter T to 12 hours for both ILC-V and HCLC-V, to ensure that they return a result within the 12-hour runtime limit. Experiments by the other algorithms that do not complete learning within the specified runtime limit are denoted as "Timeout". All experiments are based on 8GB of RAM. The experiments involving the Asia, Sports and Property networks were carried out on the Intel Core i5-8250 CPU at 1.80 GHz, whereas the experiments involving the Alarm network on the M1 CPU at 3.2 GHz.

5. Empirical results

5.1 The difference in search space explored by ILC-V and HCLC-V

This subsection investigates the difference in search space explored between the two proposed algorithms, ILC-V and HCLC-V. The comparison assumes that the PAG inputs are produced by GFCI, and relies on Step 4 (which represents the main difference between the two algorithms) where the latent confounders are assumed to be binary.

Figure 2 presents the results based on the Property network (27 nodes) for both sample sizes 1k and 10k. Figure 2a shows that ILC-V_{GFCI} produces a slightly higher p-ELBO score than HCLC-V_{GFCI}, but that ILC-V_{GFCI} achieved that by exploring considerably more search space than HCLC-V_{GFCI}; i.e., visited a total of 170 DAGs versus 20 DAGs. The charts depict different colours to illustrate how the two algorithms differ at visiting DAGs derived from MAGs that contain increasing numbers of bi-directed edges. Specifically, Figure 2a shows that ILC-V_{GFCI} visited all DAGs derived from MAGs containing up to three bi-directed

edges, whereas HCLC-V$_{\text{GFCI}}$ ended at a local maximum while visiting DAGs derived from MAGs containing up to two bi-directed edges.

Figure 2b, on the other hand, shows that the higher sample size helped ILC-V$_{\text{GFCI}}$ to both find a higher objective score and complete learning faster than HCLC-V$_{\text{GFCI}}$. This is because ILC-V$_{\text{GFCI}}$ found no DAG derived from MAGs containing two bi-directed edges to have a higher score than the highest scoring DAG derived from MAGs containing one bi-directed edge, which caused ILC-V$_{\text{GFCI}}$ to skip MAGs containing three bi-directed edges. On the other hand, HCLC-V$_{\text{GFCI}}$ ended up visiting a higher number of DAGs, but note this does not necessarily imply that the algorithm was slower; i.e., recall that HCLC-V skips checking for Markov equivalence between graphs.

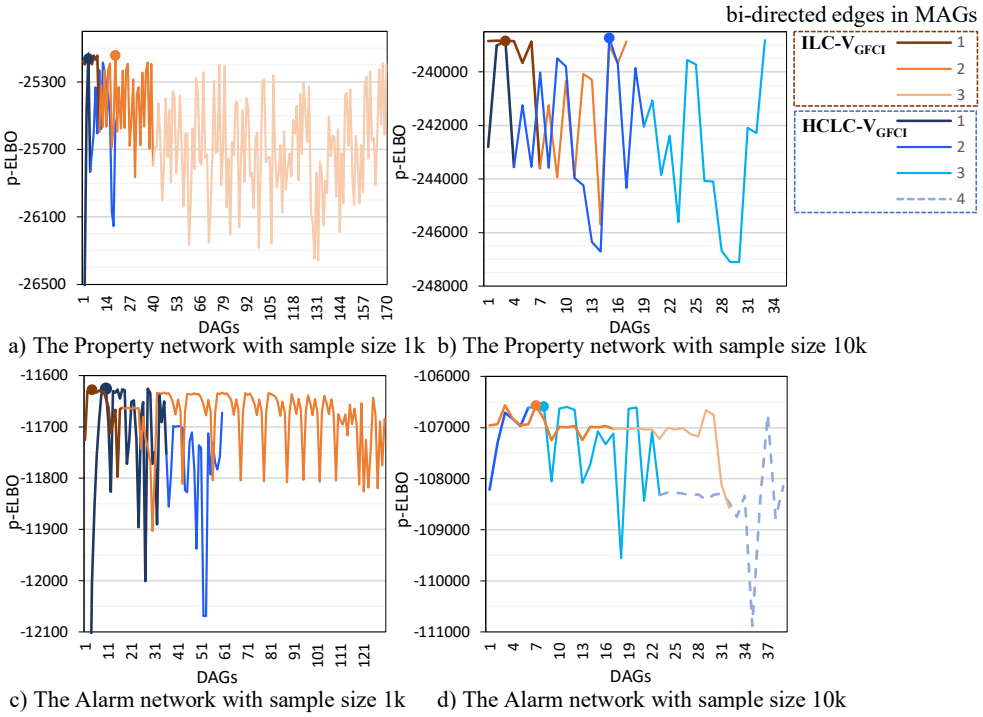

Figure 2. The p-ELBO scores produced at Step 4 by the two algorithms, where ● indicates the highest score achieved by the specified algorithm. The results in a) and b) are based on the Property network with variable 'otherPropertyExpenses' being the latent confounder and in c) and d) are based on the Alarm network with variable 'INTUBATION' being the latent confounder, and assume the input PAG is produced by GFCI.

Figure 2c and 2d repeat the analysis of Figure 2a and 2b with application to the Alarm network (37 nodes), and show that the results are consistent with those produced for the Property network. The only difference here is that, at 10k sample size, the p-ELBO score of HCLC-V$_{\text{GFCI}}$ matched that of the generally slower ILC-V$_{\text{GFCI}}$.

5.2 Performance of ILC-V and HCLC-V relative to other algorithms

We compare the results produced by ILC-V and HCLC-V to those produced by the CIL and GLSL algorithms described in subsection 2.4 which, to the best of our knowledge, are the two algorithms that are most relevant to this work, which involves both the discovery and density estimation of latent confounders.

Table 2 presents the p-ELBO score for each algorithm and data set combination, plus the p-ELBO scores of the true DAGs, for both sample sizes 1k and 10k. The average ranks show that ILC-V_{GFCI} performs best in terms of maximising the p-ELBO score across both sample sizes, followed by HCLC-V_{GFCI}. CIL algorithm is found to be the worst performing algorithm at sample size 10k, whereas GLSL mostly outperforms both ILC-V_{FCI} and HCLC-V_{FCI}, but not ILC-V_{GFCI} and HCLC-V_{GFCI}. This means that ILC-V and HCLC-V benefit from the PAG input of GFCI, and suggests that the hybrid learning GFCI might be better than FCI at recovering PAGs; an observation consistent with previous studies (Constantinou et al., 2021). Note that while the true DAG will not always have the highest p-ELBO score, the highest scores produced by the algorithms tend to be very close to those of the true DAG, and this helps to validate the results.

BN (Latent confounder)	True DAG	ILC-V_{FCI}	HCLC-V_{FCI}	ILC-V_{GFCI}	HCLC-V_{GFCI}	CIL	GLSL
	p-ELBO (sample size 1k)						
Asia (smoke)	-2258	-1845	-1845	-1807	-1807	-1796	**-1679**
Sports (Rdlevel)	-11742	**-9296**	-9417	**-9296**	-9417	-10228	-10228
Property (propertyPurchaseValue)	-25254	-34496	-34532	**-24565**	-24596	-29040	-28076
Property (borrowing)	-25254	-35042	-35080	Memory	**-24044**	-28518	-27534
Property (otherPropertyExpenses)	-25254	-35929	-35979	**-24079**	**-24079**	-29382	-28363
Alarm (INTUBATION)	-11220	Memory	-14802	**-10966**	-11068	-13777	-11581
Alarm (HYPOVOLEMIA)	-11220	Memory	-14660	**-10908**	-11010	-13721	-11117
Alarm (LVFAILURE)	-11220	Memory	-14821	**-11074**	-11075	-13989	-11307
Alarm (ERRCAUTER)	-11220	Memory	-14678	-11024	**-11017**	-13693	-11254
Alarm (PULMEMBOLUS)	-11220	Memory	-15081	**-11053**	-11055	-13994	-11294
Alarm (KINKEDTUBE)	-11220	Memory	-14948	**-10889**	-10963	-13896	-11203
Average rank		5.1	5.0	1.8	1.9	3.8	2.9
	p-ELBO (sample size 10k)						
Asia (smoke)	-22508	-17860	-17860	-17601	-17601	-17039	**-16135**
Sports (Rdlevel)	-108800	**-92014**	-92864	**-92014**	-92864	-99741	-99741
Property (propertyPurchaseValue)	-235622	-285084	-285084	**-238090**	-238267	-283142	-275212
Property (borrowing)	-235622	-277035	-277035	**-239289**	-239520	-277440	-269719
Property (otherPropertyExpenses)	-235622	-284024	-284038	-237178	**-236998**	-285975	-277949
Alarm (INTUBATION)	-105739	-119906	-119845	**-104919**	-105096	-133084	Timeout
Alarm (HYPOVOLEMIA)	-105739	Memory	-126194	**-101997**	-102960	-131819	Timeout
Alarm (LVFAILURE)	-105739	Memory	-129574	-103761	**-103720**	-134606	Timeout
Alarm (ERRCAUTER)	-105739	Memory	-121536	**-103492**	-103530	-132280	Timeout
Alarm (PULMEMBOLUS)	-105739	Memory	-126811	-103652	**-103624**	-135116	Timeout
Alarm (KINKEDTUBE)	-105739	Memory	-125698	-108480	**-102803**	-134869	Timeout
Average rank		4.4	3.7	1.5	1.8	4.4	4.2

Table 2. The p-ELBO scores for each algorithm and data set combination and across both sample sizes, where Memory indicates out-of-memory error in enumerating the possible MAGs, and Timeout indicates failure to complete learning within the 12-hour time limit. The best scores are indicated in bold.

While ILC performs best in general, it does not scale well with the size of the network. As shown in Table 2, ILC-V returns an out-of-memory error (for 8GB RAM) for most experiments with Alarm, specifically when paired with FCI, caused by the large number of possible MAGs derived from the input PAG. The cumulative runtime across all 10k sample sizes was 14, 34, 46 and 88 hours for CIL, HCLC-V_{GFCI}, ILC-V_{GFCI} and GLSL respectively, with a similar trend observed across 1k sample sizes. On average, HCLC-V is found to be 1.4 times faster than ILC-V, which in turn is found to be 1.6 times slower than CIL and 4.5 times faster than GLSL which failed to complete the Alarm network experiments at 10k sample size; suggesting that its computational efficiency might not scale well with sample size.

6. Conclusions

This work investigated two novel algorithms that can be used for both discovery and density estimation of latent confounders in BN structure learning from discrete observational data. The first algorithm (ILC-V) aims to maximise model selection accuracy by exploring sets of Markov equivalent MAGs, starting from the set of MAGs that contain the lowest number of bi-directed edges and, while the objective score increases with each set, moving to sets of MAGs with increasing numbers of bi-directed edges. The second algorithm (HCLC-V) aims to balance accuracy relative to computational efficiency by employing hill-climbing over the MAG space, enabling application to larger networks.

Both algorithms require a PAG to be provided as an input, which means that the proposed algorithms need to be paired with a structure learning algorithm that recovers ancestral graphs. Because the input PAG will typically indicate multiple possible latent confounders, the ILC-V and HCLC-V algorithms use p-ELBO as the objective function to determine the number as well as the position of the latent confounders, thereby contributing to the discovery process, in addition to density estimation, of latent confounders.

The two proposed algorithms are evaluated relative to two recent and relevant implementations that also optimise for p-ELBO. The empirical results show meaningful improvements in maximising the objective score, and in some ways in reducing time complexity; although the latter remains a major issue. Two important limitations are that a) both algorithms rely on a PAG input to be provided by some other structure learning algorithm, and b) the results are based on experiments that assume a single latent confounder only, which was necessary to ensure that most experiments complete within the 12-hour runtime limit.

References

M. Beal and Z. Ghahramani. The Variational Bayesian EM Algorithm for Incomplete Data : with Application to Scoring Graphical Model Structures. *Statistics*, 2003.

M. J. Beal and Z. Ghahramani. Variational Bayesian learning of directed graphical models with hidden variables. *Bayesian Analysis*, 1:793–831, 2006.

C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

D. Colombo and M. H. Maathuis. Order-Independent Constraint-Based Causal Structure Learning. *Journal of Machine Learning Research*, 15(116):3921–3962, 2014.

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*, 2012.

A. C. Constantinou, Y. Liu, K. Chobtham, Z. Guo, and N. K. Kitsoni. *The Bayesys data and Bayesian network repository. Queen Mary University of London, London, UK. [Online]*, 2020. URL `http://bayesian-ai.eecs.qmul.ac.uk/bayesys/`.

A. C. Constantinou, Y. Liu, K. Chobtham, Z. Guo, and N. K. Kitson. Large-scale empirical validation of Bayesian Network structure learning algorithms with noisy data. *International Journal of Approximate Reasoning*, 131:151–188, 2021. ISSN 0888-613X.

N. Friedman. The Bayesian Structural EM Algorithm. volume 98, pages 129 – 138, 01 2013.

A. Gelman, J. Carlin, H. Stern, and D. Rubin. *Bayesian Data Analysis, Second Edition*. Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis, 2003.

W. K. Hastings. Monte Carlo Sampling Methods Using Markov Chains and Their Applications. *Biometrika*, 57(1):97–109, 1970. ISSN 00063444.

M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. *An Introduction to Variational Methods for Graphical Models*, page 105–161. MIT Press, Cambridge, MA, USA, 1999.

M. Kalisch, M. Mächler, D. Colombo, M. H. Maathuis, and P. Bühlmann. Causal Inference Using Graphical Models with the R Package pcalg. *Journal of Statistical Software*, 47, 2012. doi: 10.18637/jss.v047.i11. URL `http://CRAN.R-project.org/package=pcalg`.

D. P. Kingma and M. Welling. Auto-Encoding Variational Bayes, 2013. URL `https://arxiv.org/abs/1312.6114`.

N. K. Kitson, A. C. Constantinou, Z. Guo, Y. Liu, and K. Chobtham. A survey of Bayesian Network structure learning, 2021. URL `https://arxiv.org/abs/2109.11415`.

C. Ma, S. Tschiatschek, J. M. Hernández-Lobato, R. Turner, and C. Zhang. VAEM: a Deep Generative Model for Heterogeneous Mixed Type Data, 2020.

D. Malinsky and P. Spirtes. Estimating bounds on causal effects in high-dimensional and possibly confounded systems. *International Journal of Approximate Reasoning*, 88, 2017.

J. M. Ogarrio, P. Spirtes, and J. Ramsey. A Hybrid Causal Search Algorithm for Latent Variable Models. In *Proceedings of the Eighth International Conference on Probabilistic Graphical Models*, Proceedings of Machine Learning Research, 2016.

J. Ramsey, J. Zhang, and P. Spirtes. Adjacency-Faithfulness and Conservative Causal Inference. *CoRR*, abs/1206.6843, 2012.

T. Richardson and P. Spirtes. Ancestral Graph Markov Models. *Annals of Statistics*, 2000.

F. Rodriguez-Sanchez. *mpc-mixed library*, 2021. URL `https://github.com/ferjorosa/mpc-mixed`.

F. Rodriguez-Sanchez, P. Larrañaga, and C. Bielza. Incremental Learning of Latent Forests. *IEEE Access*, 8:224420–224432, 2020. doi: 10.1109/ACCESS.2020.3027064.

F. Rodriguez-Sanchez, C. Bielza, and P. Larrañaga. Multipartition clustering of mixed data with Bayesian networks. *International Journal of Intelligent Systems*, 2022.

M. Scutari. Learning Bayesian Networks with the bnlearn R Package. *Journal of Statistical Software*, 35:1–22, 2010.

P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search, 2nd Edition*, volume 1 of *MIT Press Books*. The MIT Press, 2001.

C. Wongchokprasitti. *R-causal R Wrapper for Tetrad Library, v1.2.1*, 2019. URL `https://github.com/bd2kccd/r-causal`.