Bayesian Algorithms for Causal Data Mining

Subramani Mani SUBRAMANI.MANI@VANDERBILT.EDU
Department of Biomedical Informatics
Vanderbilt University
Nashville, TN, 37232-8340, USA

Constantin F. Aliferis CONSTANTIN.ALIFERIS@NYUMC.ORG
Center for Health Informatics and Bioinformatics
New York University
New York, NY, 10016, USA

Alexander Statnikov ALEXANDER.STATNIKOV@MED.NYU.EDU
Center for Health Informatics and Bioinformatics
New York University
New York, NY, 10016, USA

Editor: Isabelle Guyon, Dominik Janzing and Bernhard Schölkopf

Abstract

We present two Bayesian algorithms CD-B and CD-H for discovering unconfounded cause and effect relationships from observational data without assuming causal sufficiency which precludes hidden common causes for the observed variables. The CD-B algorithm first estimates the Markov blanket of a node X using a Bayesian greedy search method and then applies Bayesian scoring methods to discriminate the parents and children of X. Using the set of parents and set of children CD-B constructs a global Bayesian network and outputs the causal effects of a node X based on the identification of Y arcs. Recall that if a node X has two parent nodes A,B and a child node C such that there is no arc between A,B and A,B are not parents of C, then the arc from X to C is called a Y arc. The CD-H algorithm uses the MMPC algorithm to estimate the union of parents and children of a target node X. The subsequent steps are similar to those of CD-B. We evaluated the CD-B and CD-H algorithms empirically based on simulated data from four different Bayesian networks. We also present comparative results based on the identification of Y structures and Y arcs from the output of the PC, MMHC and FCI algorithms. The results appear promising for mining causal relationships that are unconfounded by hidden variables from observational data.

Keywords: Causal data mining, Markov blanket, Y structures

1. Introduction and Background

Causal knowledge enables us to plan interventions leading to predictable, measurable and desirable outcomes. Experimental data is typically generated for ascertaining cause and effect relationships. However, experimental studies may not be feasible in many situations due to ethical, logistical, cost, technical or other reasons. This study introduces two new Bayesian algorithms CD-B and CD-H for ascertaining causality from observational data. There are many algorithms available for learning the underlying causal structure from data such as GS (Margaritis and Thrun, 2000), PC (Spirtes et al., 2000, page 84–85), HITON (Aliferis et al., 2003a), OR (Moore and Wong, 2003) and FCI (Spirtes et al., 2000). However, all of these algorithms except FCI make an assumption of *causal sufficiency* which maintains that there are no unobserved common causes for any two or more of the observed variables. Even though FCI does not make such an assumption its usefulness is limited in practical settings due to its scalability limitation.

The Bayesian algorithms proposed in this paper are based on a computationally feasible score-based search to identify some causal effects in the large sample limit while allowing for the possibility of unobserved common causes, and without making any assumptions about the true causal structure (other than acyclicity). There is also no need to assign scores explicitly to causal structures with unobserved common causes in this framework.

We now define some terms that are needed for our causal datamining framework. Our framework for causal discovery is based on causal Bayesian networks (CBNs). A CBN is a Bayesian network in which each arc is interpreted as a direct causal influence between a parent node (variable) and a child node, relative to the other nodes in the network (Pearl, 1991). We proceed to introduce the concept of a Y structure and a Y arc in a Bayesian network. Let $W_1 \rightarrow X \leftarrow W_2$ be a V structure (there is no arc between W_1 and W_2). If there is a node Z such that there is an arc from X to Z, but no arc from W_1 to Z and no arc from W_2 to Z, then the nodes W_1, W_2, X and Z form a Y structure (see Figure 1, G_1). If such a Y structure over four measured variables $\mathbf{V}$ is learned from an observational dataset D, the arc from X to Z in the Y structure represents an unconfounded causal relationship (Mani et al., 2006). Since G_1 also has the same set of independence/dependence relationships over the observed variables (I-map) as G_{1H} (see Figure 1), the arcs $W_1 \rightarrow X$ and $W_2 \rightarrow X$ in G_1 cannot be interpreted as necessarily representing causal relationships. The arc from X to Z in a Y structure is referred to as a Y arc (YA).

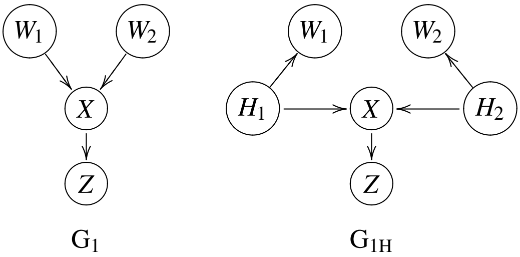


Figure 1: A Y structure G_1 and a Y equivalent structure G_{1H} (H_1 and H_2 denote hidden variables).

We now define the concept of a *Markov blanket* which is needed for an understanding of the CD-B and CD-H algorithms. The Markov blanket (MB) of a node X in a causal Bayesian network G is the union of the set of parents of X, the children of X, and the parents of the children of X.

2. Algorithms

In this section we introduce the algorithms in this study for discovering cause and effect relationships from observational data. We first introduce the Bayesian algorithms CD-B and CD-H that learn global CBN models and output the set of Y arcs which represent cause and effect relationships unconfounded by hidden variables. We then provide short descriptions of the

PC, FCI and MMHC algorithms and the post-processing procedure that we use to identify the unconfounded causal arcs from the output of PC and MMHC.

2.1 CD-B algorithm

The CD-B algorithm first induces the Markov blanket (MB) of each node $X \in \mathbf{V}$ (where $\mathbf{V}$ is the set of domain variables) using the Bayesian Markov blanket induction (MBI) procedure (Mani, 2005). The MBI procedure finds the Markov blanket of a node X under the assumptions of Markov, faithfulness and large sample size. It uses a greedy forward and backward search in seeking the Markov blanket of X, which we denote $\mathrm{MB}(X)$. The set $\mathrm{MB}(X)$ is the estimated Markov blanket of X in a data generating network. From the MB of each node X the spouse nodes (parents of children of a node X are referred to as the spouse nodes of X) are excluded by a Bayesian dependence heuristic (Cooper, 1997) to obtain the set of parents and children of X (denoted as $\mathrm{PC}(X)$). Using $\mathrm{PC}(X)$ we generate all possible DAGs such that the only arcs are from each parent to X and from X to each child. We refer to these DAGs as PC DAGs.

The key insight is that there are exactly 2^k such PC DAGs where $k = |\mathrm{PC}(X)|$. The highest scoring DAG from each node set $\mathrm{PC}(X) \cup X$ is used to get the $\mathbf{P}(X)$ and the $\mathbf{C}(X)$, that is, the parents and children of X respectively. Using $\mathbf{P}(X)$ from the highest scoring PC DAGs with two or more parents a global directed graph is constructed. Note that a PC DAG G with $\mathbf{P}(X)$ as set of parents and $\mathbf{C}(X)$ as set of children of the node X is unique (the only member of its Markov equivalence class) if $|\mathbf{P}(X)| \geq 2$. Since all the PC DAGs are scored, this step is exponential in the size of the set of parents and children. To the best of our knowledge the PC DAG method introduced here is the only Bayesian method to partition a set of parents and children ($\mathbf{P}(X) \cup \mathbf{C}(X)$) into the set of parents $\mathbf{P}(X)$ and the set of children $\mathbf{C}(X)$.

The global directed graph created using the set of parents may contain directed cycles. A directed cycle is a directed path starting from a node A and ends in node A after traversing two or more nodes. The cycles in the graph are broken iteratively by removing the "weakest" arc using a greedy search heuristic till all cycles are eliminated. The $\mathbf{C}(X)$ edges from the highest scoring PC DAGs with two or more parents are inserted based on a set of constraints (rules). The union of the edges of the highest scoring PC DAGs with less than two parents are inserted using a different set of constraints. As already mentioned, when the PC DAG has two or more parents it is unique, that is, it is the only member of its Markov equivalence class. On the other hand the PC DAGs with less than two parents are not unique (there is at least one additional member in its Markov equivalence class). Hence the arcs belonging to the two categories of PC DAGs are inserted into the global DAG using different sets of constraints. The resulting DAG is used to identify all the Y arcs. The pseudocode for the CD-B algorithm is provided in Appendix A.1.

2.2 CD-H algorithm

The CD-H algorithm replaces the initial steps of the CD-B algorithm for finding the $\mathrm{PC}(X)$ with the MMPC algorithm (Tsamardinos et al., 2003, 2006). The MMPC uses a two-phase search procedure based on tests of independence/dependence. In the first phase of search a candidate set of parents and children called CPC is estimated which is a superset of the parents and children (PC) set. The second phase of the search procedure prunes the CPC set yielding the PC set. A proof of correctness and empirical results showing the validity of the MMPC algorithm are provided in (Tsamardinos et al., 2003). The subsequent steps of the CD-H algorithm are similar to CD-B.

2.3 PC algorithm

The PC algorithm takes as input a dataset D over a set of observed random variables **V**, a conditional independence test, and an α level of significance threshold for a test of statistical independence and then outputs an essential graph. PC also makes an assumption of *causal sufficiency*. This means that all the variables of the causal network are measured and there is no attempt to discover latent (hidden) variables. Hence PC is not designed to discover hidden variables that are common causes of any pair of observed variables. In the worst case, PC is exponential in the largest degree (size of the set of parents and children of a node) in the data generating DAG. See (Spirtes et al., 2000, page 84–85) for more details on the PC algorithm. The PC algorithm outputs both directed and undirected edges. A post-processing step (procedure YA) that we add is performed on the set of arcs to identify the Y structures. The pseudocode for procedure YA is given in Appendix A.1.3.

2.4 FCI algorithm

The FCI algorithm takes as input a dataset D over a set of random variables **V** and outputs a graphical model consisting of edges between variables that have a cause and effect interpretation. While the PC algorithm outputs only directed and undirected edges, the FCI algorithm outputs a richer set of edges to denote the presence of hidden (unmeasured) confounding variables and various levels of uncertainty in the orientation of the edges (Spirtes et al., 2000). The FCI algorithm can handle hidden variables and sample selection bias that are likely to be present in real-world datasets. It is possible to obtain causal relationships that are unconfounded by hidden variables from the partial ancestral graph (PAG) output of the FCI algorithm. The edges oriented as $A \rightarrow B$ in the FCI output can be interpreted as an unconfounded causal arc similar to a Y arc.

2.5 MMHC algorithm

The max-min hill-climbing (MMHC) Bayesian network structure learning algorithm is a hybrid algorithm that combines ideas from constraint-based and score-based methods (Tsamardinos et al., 2006). MMHC has been extensively evaluated on a variety of structure learning tasks from different datasets and outperformed PC, FCI, the Sparse Candidate, Optimal Reinsertion and the Greedy Equivalence Search algorithms. The MMHC algorithm estimates the set of parents and children of a node X denoted by PC(X) using the MMPC algorithm (Tsamardinos et al., 2003) to first obtain an undirected skeleton of the output graph. MMHC then uses greedy steepest-ascent TABU search and the Bayesian scoring measure BDeu (Heckerman et al., 1995) to orient the edges.

3. Experimental methods

In this section we describe the experimental methods used to evaluate our causal discovery approach. We used expert-defined CBNs to (1) generate data from those models, (2) apply the causal discovery algorithm to the data, and (3) evaluate the causal relationships output by the algorithm relative to the data generating CBNs that serve as gold standards. The output of the algorithm was compared with the data generating structure and scored as explained below. CD-B and CD-H algorithms were implemented in Matlab. The PC and FCI algorithms implemented in Tetrad IV (`http://www.phil.cmu.edu/projects/tetrad`) were used. The MMHC implementation in the Causal Explorer package (Aliferis et al., 2003b) was used. For PC, MMHC, CD-B and CD-H algorithms the Y arcs output by the algorithms were compared with the Y arcs of the data generating networks and for FCI the fully oriented arcs were used. Recall that a post-processing step was required for PC and MMHC algorithms to obtain the Y arcs. Precision, recall and F-measure were computed for the algorithms as follows:

Precision: (# of Y arcs correctly identified) / (# of total Y arcs output).

Recall: (# of Y arcs correctly identified) / (# of total Y arcs present in the data generating network).

F measure: (2 * recall * precision) / (recall + precision).

Four Bayesian networks built by domain experts in such varied fields as medicine, atmospheric sciences and agriculture were identified. These networks are Alarm (Beinlich et al., 1990), Hailfinder (Abramson et al., 1996), Barley (Kristensen and Rasmussen, 2002), and Munin (Andreassen et al., 1987). For causal discovery, we generated simulated training instances by stochastic sampling (Henrion, 1986). Varying sample sizes in the range of 1,000 to 20,000 instances were used in our causal discovery experiments. Table 1 gives the distribution of the nodes, arcs and Y structures for the various networks used in our study. Typically default parameters were used to run the algorithms with some adjustments made for uniformity. The PC algorithm was run with default parameters (significance level 0.05). CD-B was also run with default parameters (maximum MB size 12, dependency threshold for spouse elimination 0.9). CD-H was run with the following parameters: maximum size of conditioning set 10 and significance threshold 0.05. All the algorithms were run on each of the sample sizes using the ACCRE (Linux) cluster in Vanderbilt University consisting of x86 processors with 3.8 GB memory. Each job was assigned to a single processor with a time limit of 48 hours.

Table 1: Nodes, arcs and Y structures in the Alarm, Hailfinder, Barley, and Munin networks

Category	Alarm	Hailfinder	Barley	Munin
Nodes	37	56	48	189
Arcs	46	66	84	282
Y structures	13	20	44	147

4. Results

The results presented below are based on sample sizes of 1K, 2K, 5K, 10K and 20K instances for each of the four domain datasets that were generated. We present a summary performance of all the four algorithms based on Y arcs present in all the data generating networks using precision, recall and F-measure as explained below. The aggregate results are presented based on the following two methods.

(i). The various data generating networks are given equal weight in the analysis irrespective of the number of Y arcs.

(ii). The data generating networks are weighted by the number of Y arcs present in each network.

Altogether there were 224 YA in the four domain CBNs. The results presented are based on averages over all the four networks unless specified otherwise (see Tables 2, 3 and Figures 2, 3). The highest precision of 0.97 (equal weight) and 0.95 (weighted by # of Y arcs) were achieved at

Table 2: Averages without FCI table weighted by # of Y arcs.

F-measure

Sample	CD-B	CD-H	PC	MMHC
1k	0.34	0.27	0.15	0.32
2k	0.41	0.30	0.22	0.33
5k	0.47	0.34	0.29	0.34
10k	0.47	0.38	0.30	0.52
20k	0.52	0.44	0.34	0.44

Precision

Sample	CD-B	CD-H	PC	MMHC
1k	0.79	0.28	0.90	0.43
2k	0.84	0.31	0.93	0.40
5k	0.82	0.31	0.95	0.40
10k	0.83	0.39	0.79	0.57
20k	0.81	0.39	0.65	0.45

Recall

Sample	CD-B	CD-H	PC	MMHC
1k	0.21	0.27	0.08	0.25
2k	0.27	0.29	0.13	0.29
5k	0.33	0.38	0.17	0.29
10k	0.33	0.38	0.18	0.48
20k	0.38	0.50	0.23	0.44

Table 3: Averages without FCI table weighted equally.

F-measure

Sample	CD-B	CD-H	PC	MMHC
1k	0.32	0.41	0.20	0.40
2k	0.39	0.50	0.35	0.46
5k	0.49	0.51	0.45	0.38
10k	0.48	0.52	0.38	0.60
20k	0.54	0.55	0.45	0.58

Precision

Sample	CD-B	CD-H	PC	MMHC
1k	0.76	0.44	0.93	0.61
2k	0.78	0.52	0.93	0.59
5k	0.76	0.51	0.97	0.51
10k	0.76	0.54	0.78	0.69
20k	0.80	0.59	0.73	0.60

Recall

Sample	CD-B	CD-H	PC	MMHC
1k	0.29	0.39	0.14	0.32
2k	0.32	0.49	0.26	0.40
5k	0.39	0.57	0.36	0.32
10k	0.39	0.55	0.33	0.56
20k	0.44	0.57	0.43	0.58

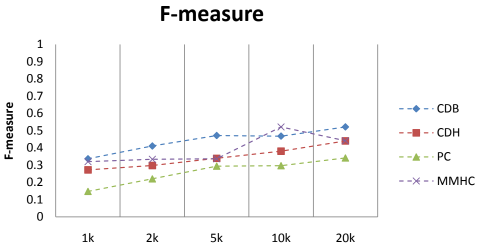


Figure 2: Averages without FCI graph weighted by # of Y arcs.

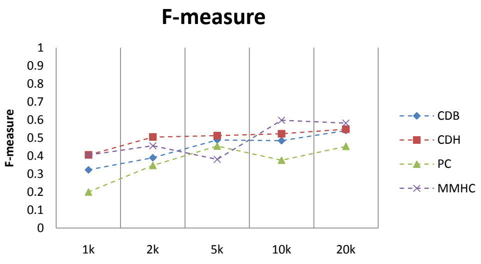
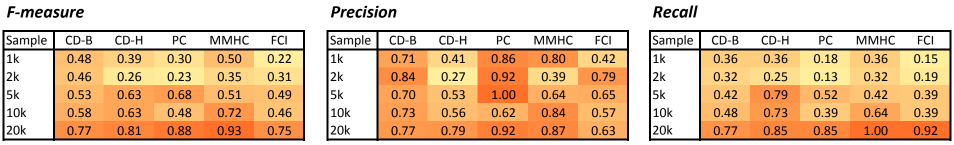


Figure 3: Averages without FCI graph weighted equally for all networks.

the sample size of 5,000 with PC. In general PC and CD-B had higher precision (≥ 0.65) across all the sample sizes tested. The best recall was obtained by MMHC (0.58 equally weighted) and CD-H (0.50 when weighted by # of Y arcs) with a sample size of 20,000. The best F-measure (when equally weighted) of 0.60 was achieved by MMHC (sample size 10K) followed by 0.55 for CD-H (sample size 20K). The best F-measure (weighted by # of Y arcs) of 0.52 was achieved by MMHC (sample size 10K) and CD-B (sample size 20K).

FCI could be run without going out of memory or exceeding the time limit of 48 hours on all sample sizes only for the Alarm dataset. Out of 20 experiments (5 sample sizes x 4 datasets), FCI ran out of memory in 9 cases (for all sample sizes in Barley network and for sample sizes 1K, 5K, 10K, and 20K in Munin network), ran out of time in 1 case (sample size 20K for Hailfinder network), and completed with results in the remaining 10 experiments (see Tables 4, 5 and Figures 4, 5). Based on F measure FCI performance is generally lower when compared with the other algorithms across all the sample sizes. Figure 6 shows total run time for all algorithms for the latter 10 experiments. As can be seen, FCI is the second slowest algorithm after CD-H. However, CD-H was able to complete with results in all 20 experiments, thus it is more useful for practitioners despite being one of the slowest algorithms in the comparison. CD-H runs slow primarily because it includes false positives in the estimated PC sets which makes PC DAG search much more computationally expensive. Table 6 provides the runtimes of the various algorithms for the Alarm dataset. FCI runtime is an order of magnitude higher compared to the other algorithms on the Alarm dataset.

Table 4: Averages with FCI table weighted by # of Y arcs.



F-measure

Sample	CD-B	CD-H	PC	MMHC	FCI
1k	0.48	0.39	0.30	0.50	0.22
2k	0.46	0.26	0.23	0.35	0.31
5k	0.53	0.63	0.68	0.51	0.49
10k	0.58	0.63	0.48	0.72	0.46
20k	0.77	0.81	0.88	0.93	0.75

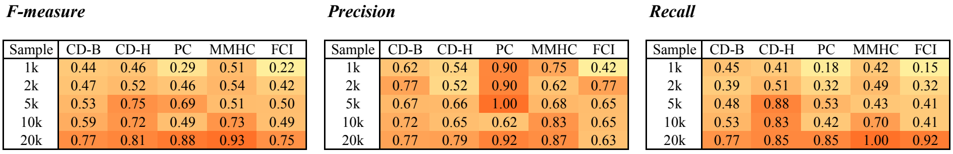
Precision

Sample	CD-B	CD-H	PC	MMHC	FCI
1k	0.71	0.41	0.86	0.80	0.42
2k	0.84	0.27	0.92	0.39	0.79
5k	0.70	0.53	1.00	0.64	0.65
10k	0.73	0.56	0.62	0.84	0.57
20k	0.77	0.79	0.92	0.87	0.63

Recall

Sample	CD-B	CD-H	PC	MMHC	FCI
1k	0.36	0.36	0.18	0.36	0.15
2k	0.32	0.25	0.13	0.32	0.19
5k	0.42	0.79	0.52	0.42	0.39
10k	0.48	0.73	0.39	0.64	0.39
20k	0.77	0.85	0.85	1.00	0.92

Table 5: Averages with FCI table weighted equally.



F-measure

Sample	CD-B	CD-H	PC	MMHC	FCI
1k	0.44	0.46	0.29	0.51	0.22
2k	0.47	0.52	0.46	0.54	0.42
5k	0.53	0.75	0.69	0.51	0.50
10k	0.59	0.72	0.49	0.73	0.49
20k	0.77	0.81	0.88	0.93	0.75

Precision

Sample	CD-B	CD-H	PC	MMHC	FCI
1k	0.62	0.54	0.90	0.75	0.42
2k	0.77	0.52	0.90	0.62	0.77
5k	0.67	0.66	1.00	0.68	0.65
10k	0.72	0.65	0.62	0.83	0.65
20k	0.77	0.79	0.92	0.87	0.63

Recall

Sample	CD-B	CD-H	PC	MMHC	FCI
1k	0.45	0.41	0.18	0.42	0.15
2k	0.39	0.51	0.32	0.49	0.32
5k	0.48	0.88	0.53	0.43	0.41
10k	0.53	0.83	0.42	0.70	0.41
20k	0.77	0.85	0.85	1.00	0.92

Table 6: Alarm original network runtimes in minutes for all the algorithms.

Sample	CD-B	CD-H	PC	MMHC	FCI
1k	0.20	0.30	0.10	0.10	5.00
2k	0.30	0.30	0.10	0.10	1.00
5k	0.50	0.50	0.10	0.20	5.00
10k	0.70	0.70	0.10	0.20	5.00
20k	1.10	0.90	0.20	0.40	58.00

We also present results of causal discovery in the presence of hidden variables based on randomly assigning “hidden” status to 25% of the variables for the Alarm dataset (see Tables 7

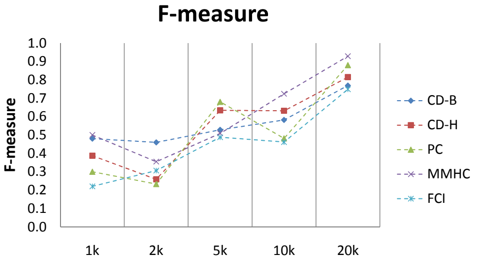


Figure 4: Averages with FCI graph weighted by # of Y arcs.

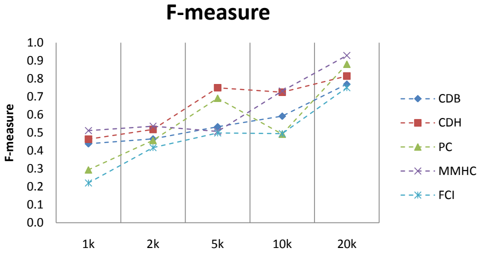


Figure 5: Averages with FCI graph weighted equally for all networks.

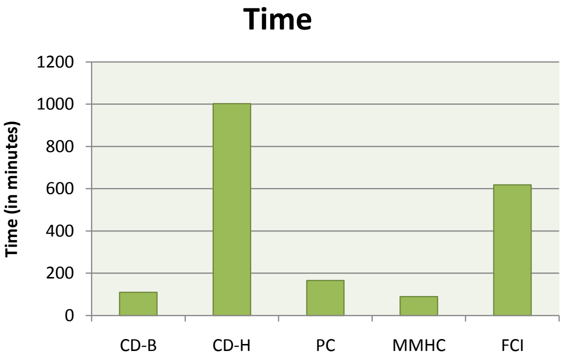
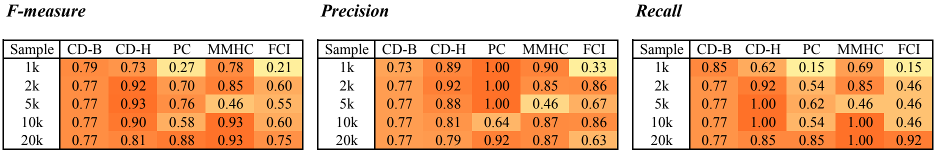


Figure 6: Runtimes for all the algorithms over all the four datasets based on FCI completion.

and 8). The results presented in Table 8 are averaged over 5 such random Alarm networks with with 25% of the nodes hidden. The results show that there is a degradation in performance for all the algorithms when a subset of the variables are unobserved (see Table 9). CD-B and FCI appear more robust in the presence of hidden variables when compared to CD-H, PC and MMHC based on the magnitude of reduction in F measure when hidden variables are introduced. Additional evaluation is needed to understand the effect of hidden variables for causal discovery from observational data.

Table 7: Alarm original.



F-measure

Sample	CD-B	CD-H	PC	MMHC	FCI
1k	0.79	0.73	0.27	0.78	0.21
2k	0.77	0.92	0.70	0.85	0.60
5k	0.77	0.93	0.76	0.46	0.55
10k	0.77	0.90	0.58	0.93	0.60
20k	0.77	0.81	0.88	0.93	0.75

Precision

Sample	CD-B	CD-H	PC	MMHC	FCI
1k	0.73	0.89	1.00	0.90	0.33
2k	0.77	0.92	1.00	0.85	0.86
5k	0.77	0.88	1.00	0.46	0.67
10k	0.77	0.81	0.64	0.87	0.86
20k	0.77	0.79	0.92	0.87	0.63

Recall

Sample	CD-B	CD-H	PC	MMHC	FCI
1k	0.85	0.62	0.15	0.69	0.15
2k	0.77	0.92	0.54	0.85	0.46
5k	0.77	1.00	0.62	0.46	0.46
10k	0.77	1.00	0.54	1.00	0.46
20k	0.77	0.85	0.85	1.00	0.92

5. Discussion

In this section we discuss the results and present the implications of our research for discovering causal relationships from observational data. This research has highlighted the role of Y structures for causal discovery from observational data and introduced two new algorithms CD-B and CD-H based on identification of Y structures.

Table 8: Alarm 75 percent observed.

F-measure

Sample	CD-B	CD-H	PC	MMHC	FCI
1k	0.60	0.38	0.22	0.49	0.19
2k	0.60	0.50	0.32	0.43	0.40
5k	0.51	0.65	0.49	0.27	0.43
10k	0.57	0.54	0.54	0.47	0.58
20k	0.58	0.61	0.56	0.57	0.66

Precision

Sample	CD-B	CD-H	PC	MMHC	FCI
1k	0.75	0.53	0.70	0.75	0.28
2k	0.73	0.57	0.80	0.52	0.41
5k	0.67	0.65	1.00	0.38	0.51
10k	0.64	0.58	0.69	0.53	0.51
20k	0.65	0.60	0.80	0.66	0.58

Recall

Sample	CD-B	CD-H	PC	MMHC	FCI
1k	0.52	0.31	0.13	0.38	0.15
2k	0.52	0.46	0.21	0.38	0.39
5k	0.42	0.67	0.35	0.22	0.38
10k	0.52	0.52	0.45	0.44	0.67
20k	0.54	0.62	0.47	0.54	0.79

Table 9: Alarm 75 performance degradation.

Sample	CD-B	CD-H	PC	MMHC	FCI
1k	-0.18	-0.35	-0.04	-0.30	-0.02
2k	-0.17	-0.43	-0.38	-0.42	-0.20
5k	-0.26	-0.28	-0.27	-0.19	-0.11
10k	-0.20	-0.36	-0.05	-0.45	-0.02
20k	-0.18	-0.21	-0.32	-0.36	-0.09

Precision varied within a narrow range of 0.76 to 0.84 for CD-B and between 0.65 and 0.97 for PC (see Tables 2 and 3). The relatively narrow precision range for the different sample sizes combined with a monotonic increase in recall throughout the sample range shows that the performance of CD-B and PC is robust across a wide range of sample sizes. In general precision values are higher compared to recall values for all the sample sizes except for the CD-H algorithm. Note that a higher precision translates to lower number of false positives even though some causal relationships may not be reported. A desirable goal in causal discovery is to keep the proportion of false positives low even if it entails a trade-off in terms of recall.

FCI and CD-H had longer runtimes when compared with PC, MMHC and CD-B. It is possible to use symmetry correction in the MMPC step of the CD-H algorithm to reduce the number of false positives in the PC set and decrease runtime.

The causal discovery framework that we presented for identifying direct causal relationships is dependent on the presence of Y structures in the data generating process. The two medical (Alarm, Munin) and two non-medical (Hailfinder, Barley) networks that were used to generate data had varying numbers of Y structures. These networks were created by domain experts capturing the probabilistic dependencies and independencies in the domain. Hence it seems plausible that Y structures occur in the data generating process of many real-world domains. Presence of Y structures have also been shown in a real world infant birth and death dataset (Mani and Cooper, 2004).

CD-B and CD-H are unique in differentiating the set of parents and the set of children of a node X from the union of the set of parents and children of X. Identification of the parents and children of a node will give us the candidate set of direct causes and the candidate set of direct effects of a node. Due to the presence of hidden variables all the parents cannot be interpreted as direct causes and all the children cannot be interpreted as direct effects. However, the candidate set of parents and children can be used to rule out hypothesized causes or effects. Also, when experimental studies are feasible the candidate sets can act as the first filter and provide the experimenter with a preliminary set of potential causes and effects. Moreover, the set of parents or the set of children of a node completely specify a directed acyclic graph which can be used to approximate the data generating model.

5.1 Related work

The most related algorithm to the CD-B algorithm is the BLCD (Mani and Cooper, 2004; Mani, 2005). BLCD estimates the Markov blanket of a variable and uses it for the identification of Y structures from sets of four variables. BLCD does not specifically identify the sets of parents and children from the Markov blanket.

Aliferis et al. have introduced HITON, an algorithm to determine the MB of an outcome variable (Aliferis et al., 2003a). Tsamardinos et al. have described an algorithm called MMMB and they discuss that since the MB contains direct causes and direct effects of a variable X, the MB has causal interpretability (Tsamardinos et al., 2003). Note that both HITON and MMMB do not specifically distinguish between causes and effects of a node; however, they do output the variables that have direct edges during the operation of the algorithm. Additional processing (or experimentation) of HITON and MMMB output is required to determine causal directionality.

5.2 Limitations and future work

There are two main types of limitations of this work. The first set of limitations results from the framework and assumptions we have chosen for causal discovery. The second set of limitations is due to the specifics of the algorithm and the experimental methods that were used.

The CBN framework imposes a directed acyclic graph structure on all causal phenomena. Discovering causal mechanisms that incorporate feedback cycles can be problematic unless time is represented explicitly and cycles are "unfolded" to provide a DAG structure (Cooper, 1999). The causal discovery approach we have taken is not complete in the sense that we can discover only causal relationships represented in nature as Y structures. The algorithms also currently requires that the modeled variables be discrete.

The evaluation measures of precision, recall and F measure that were used are structural. Hence the evaluation of the purported causal relationships were structural, leaving out the parametric components. That is, we evaluated how well the algorithm can discover the presence of a causal influence, but leave to future work the characterization of how well the algorithm captures the functional relationships among the causes and effects.

We plan to apply the CD-B and CD-H algorithms to real-world datasets as part of our future work.

Acknowledgments

We thank professor Greg Cooper for helpful discussions. We thank Yerbolat Dosbayev for implementing CD-B and CD-H and Yukun Chen for running the experiments and generating the results presented in the paper. We also thank the anonymous reviewers for their critical comments and suggestions for improving the paper.

References

Bruce Abramson, John Brown, Ward Edwards, Allan Murphy, and Robert L. Winkler. Hailfinder: A Bayesian System for Forecasting Severe Weather. *International Journal of Forecasting*, 12:57–71, 1996.

Constantin F. Aliferis, Ioannis Tsamardinos, and Alexander Stanikov. HITON, A novel markov blanket algorithm for optimal variable selection. In *Proceedings of the AMIA Fall Symposium*, 2003a.

Constantin F. Aliferis, Ioannis Tsamardinos, Alexander Stanikov, and Laura E. Brown. Causal Explorer: A causal probabilistic network learning toolkit for biomedical discovery. In *Proceedings of the 2003 International Conference on Mathematics and Engineering Techniques in Medicine and Biological Sciences (METMBS)*, 2003b.

Steen Andreassen, Marianne Woldbye, Bjorn Falck, and Stig K. Andersen. MUNIN — A causal probabilistic network for interpretation of electromyographic findings. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, pages 366–372, San Mateo, CA, 1987. Morgan Kaufmann.

Ingo A. Beinlich, H.J. Suermondt, R. Martin Chavez, and Gregory F. Cooper. The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. In *Proceedings of the Second European Conference on Artificial Intelligence in Medicine*, pages 247–256, London, 1990. Chapman and Hall.

Gregory F. Cooper. A simple constraint-based algorithm for efficiently mining observational databases for causal relationships. *Data Mining and Knowledge Discovery*, 1:203–224, 1997.

Gregory F. Cooper. An Overview of the Representation and Discovery of Causal Relationships Using Bayesian Networks. In Clark Glymour and Gregory F. Cooper, editors, *Computation, Causation, and Discovery*, pages 3–62. MIT Press, Cambridge, MA, 1999.

David Heckerman, Dan Geiger, and David M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20(3):197–243, 1995.

Max Henrion. Propagating uncertainty in bayesian networks by probabilistic logic sampling. In *Proceedings of the 2nd Annual Conference on Uncertainty in Artificial Intelligence (UAI-86)*, New York, NY, 1986. Elsevier Science Publishing Company, Inc.

K. Kristensen and I.A. Rasmussen. The use of a Bayesian network in the design of a decision support system for growing malting barley without use of pesticides. *Computers and Electronics in Agriculture*, 33:197–217, 2002.

Subramani Mani. *A Bayesian Local Causal Discovery Framework*. PhD thesis, University of Pittsburgh, 2005.

Subramani Mani and Gregory F. Cooper. Causal discovery using a Bayesian local causal discovery algorithm. In M. Fieschi et al. editor, *Proceedings of MedInfo*, pages 731–735. IOS Press, 2004.

Subramani Mani, Peter Spirtes, and Gregory F. Cooper. A theoretical study of Y structures for causal discovery. In Rina Dechter and Thomas S. Richardson, editors, *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 314–323, Corvallis, OR, 2006. AUAI Press.

Dimitris Margaritis and Sebastian Thrun. Bayesian network induction via local neighborhoods. In S.A.Solla, T.K.Leen, and K.R.Muller, editors, *Advances in neural information processing systems*, volume 12, pages 505–511, Cambridge, MA, 2000. MIT Press.

Andrew Moore and Weng-Keen Wong. Optimal reinsertion: A new search operator for accelerated and more accurate bayesian network structure learning. In T. Fawcett and N. Mishra, editors, *Proceedings of the 20th International Conference on Machine Learning (ICML '03)*, pages 552–559, Menlo Park, California, August 2003. AAAI Press.

Judea Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, San Francisco, California, 2nd edition, 1991.

Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search*. MIT Press, Cambridge, MA, 2nd edition, 2000.

Ioannis Tsamardinos, Constantin F. Aliferis, and Alexander Stanikov. Time and sample efficient discovery of markov blankets and direct causal relations. In *Proceedings of the 9th CAN SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 673–678, 2003.

Ioannis Tsamardinos, Laura Brown, and Constantin Aliferis. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, 65:31–78, 2006.

Appendix A. CD-B Pseudocode

In this section we provide the pseudocode for the CD-B algorithm and the details of the various procedures called by CD-B, specifically the Markov blanket induction (MBI) procedure and the Y arc (YA) finding procedure.

A.1 CD-B algorithm

/* Note: When the PC DAG has 0 or 1 parent G_{max} is not unique. We pick any G_{max} PC DAG from its equivalence class. This implies that in the data generating DAG the edges of such PC DAGs can have either $A \rightarrow B$ or $A \leftarrow B$ orientation. */

Input : Dataset D and the set of variables $\mathbf{X}$.

Output : Pairwise causal influences of the form $A \rightarrow B$ representing Y arcs.

The following are the steps of the algorithm:

(i). For each variable $X \in \mathbf{X}$ estimate MB(X) using the Bayesian MB induction (MBI) procedure.

(ii). For each variable $X \in \mathbf{X}$ DO

- (a) Update MB(X). If A is in the MB of B, but B is not in the MB of A, we add B to the MB of A.
- (b) Remove the spouse nodes from MB(X) to obtain PC(X). Any node independent of X is excluded from MB(X). Let $\mathbf{B}$ denote PC(X).
- (c) From $\mathbf{B} \cup X$ generate all possible DAGs such that the only arcs are from each parent to X and from X to each child. Let this set of DAGs be $\mathbf{G}$.
- (d) From the set of DAGs $\mathbf{G}$ identify the maximally scoring DAG G using the BDeu scoring measure (Heckerman et al., 1995). Let this DAG be G_{max}. If there is a tie for G_{max}, it is broken randomly.
- (e) If the G_{max} has 2 or more parents mark the Pa(X) and Ch(X) as oriented ($\mathrm{Pa}^o(X)$ and $\mathrm{Ch}^o(X)$).
- (f) If the G_{max} has less than 2 parents mark the Pa(X) and Ch(X) as unoriented ($\mathrm{Pa}^u(X)$ and $\mathrm{Ch}^u(X)$).

(g) OD

(iii). Using $\mathrm{Pa}^o(X)$ of all the nodes with 2 or more parents construct a global directed graph G'. G' may contain cycles.

(iv). Construct DAG G from G' using Procedure RC.

(v). Let **E** be the union of all the arcs from $\mathrm{Ch}^o(X)$ of all the nodes with 2 or more parents.

(vi). While **E** not $\emptyset$, insert the edge $e \in \mathbf{E}$ in G iff it satisfies the following conditions (a), (b) and (c).

(a) Not already present in G.

(b) No cycle is introduced in G.

(c) Insert the *e* that maximizes the score for G.

(d) Remove *e* from **E**.

(e) OD

(vii). Let **E** be the union of all the edges ignoring direction from $\mathrm{Pa}^u(X)$ and $\mathrm{Ch}^u(X)$ of all the nodes with less than 2 parents.

(viii). While **E** not $\emptyset$, insert the edge $e \in \mathbf{E}$ ($A \rightarrow B$ or $A \leftarrow B$) in G iff it satisfies the following conditions (a), (b) , (c) and (d).

(a) Not already present in G ignoring direction.

(b) No new V structure is introduced in G.

(c) No cycle is introduced in G.

(d) Insert the *e* that maximizes the score for G.

(e) Remove *e* from **E**.

(f) OD

(ix). Remove cycles from G using Procedure RC.

(x). Identify all the Y arcs (YA) in G using Procedure YA and output the YA.

A.1.1 PROCEDURE MBI

We derive an estimate of the Market blanket (MB) of a node (designated as **H**) using a greedy forward and backward heuristic search which we refer to as the *Procedure MBI.*

Input: Dataset D over observed random variables **X** and a variable $X \in \mathbf{X}$.

Output: Markov blanket of X in a data generating network, which we denote MB(X) i.e. the union of estimated parents, children and spouses (parents of children) of node X, under the assumption the data is being generated by a faithful Bayesian network on measured variables **X**.

The following are the steps of the MBI procedure:

- Identify the set $\mathbf{H'} \subseteq \mathbf{X} \backslash X$ that maximizes the BDeu score for the structure $\mathbf{H'} \rightarrow X$ based on a one-step forward greedy search.

- Perform a one step backward greedy search that prunes **H'** to yield set $\mathbf{H} \subseteq \mathbf{H'}$ that maximizes the score for the structure $\mathbf{H} \rightarrow X$.

- Output **H** which represents MB(X).

A.1.2 PROCEDURE RC

This procedure removes the cycles from a directed graph. The "weakest" arc is removed iteratively till all cycles are eliminated.

Input: A directed graph G'.

Output: A directed acyclic graph G.

The following are the steps of the procedure:

(i). Check for cycle(s) in G'. If no cycle assign G' to G and return G.

(ii). Identify all the arcs forming cycle(s). Let these set of arcs be **E**.

(iii). Identify the weakest arc E $\in$ **E** by iteratively removing each arc from **E** and scoring the graph using the BDeu scoring measure. The arc causing the least reduction in the BDeu score is determined to be the weakest.

(iv). Remove E from G'. Let the resulting graph be G'. GOTO Step 1.

A.1.3 PROCEDURE YA

We identify all the unique Y arcs (YA) in a DAG G using this procedure. The procedure looks for all the embedded Y structures (EYS) in G. We say that G contains an *embedded* Y structure involving the variables W_1, W_2, X and Z, iff all and only the following adjacencies hold among the variables W_1, W_2, X and Z ($A \square B$ means that there is no arc between A and B):

- $W_1 \square W_2$; $W_1 \square Z$; $W_2 \square Z$
- $W_1 \rightarrow X$; $W_2 \rightarrow X$; $X \rightarrow Z$

Input: A DAG G and a set of nodes **X** in G.

Output: A set of Y arcs denoted as **Y**.

```
Initialize set of YA as $\mathbf{Y} := \{\}$.
For each $X \in \mathbf{X}$
DO
   Determine Pa($X$) for $X$.
   If $|\mathrm{Pa}(X)| \leq 1$
      Continue /* Next iteration */
   Determine Ch($X$) for $X$.
   If $|\mathrm{Ch}(X)| < 1$
      Continue /* Next iteration */

   /* Look for Y structure */
   For each pair of parents $W_1, W_2$ of $X$
   DO
      If $W_1$ and $W_2$ are adjacent then Continue
      For each child $Z \in \mathrm{Ch}(X)$
      DO
         If $(W_1, Z)$ or $(W_2, Z)$ adjacent then Continue
         If $(X \rightarrow Z) \notin \mathbf{Y}$
```

```
            $\mathbf{Y} := \mathbf{Y} \cup \{X \to Z\}$
        OD
    OD
OD
Return $\mathbf{Y}$
```