Partial Correlation- and Regression-Based Approaches to Causal Structure Learning

Technical Report

Jean-Philippe Pellet[1,2] and André Elisseeff[1]

[1] IBM Research
Business Optimization Group
Säumerstr. 4, 8803 Rüschlikon, Switzerland
{jep,ael}@zurich.ibm.com

[2] Swiss Federal Institute of Technology Zurich
Machine Learning Group
Institute of Computational Science
Universitätstr. 6, 8092 Zurich, Switzerland

Abstract. We present the Total Conditioning (TC) algorithm for causal discovery suited in the presence of continuous variables. Given a set of n data points drawn from a distribution whose underlying causal structure is a directed acyclic graph (DAG), the TC algorithm returns a structure, i.e., a DAG, over the variables that tends to the correct structure when n tends to infinity. The approach builds on the structural equation modeling framework, well suited for continuous variables, and relies on causal Bayesian networks semantics to handle consistency. We compare TC and a variant, TC_{bw}, which borrows techniques from feature selection for robustness when the number of samples is small, to the state-of-the-art PC algorithm. We show that TC_{bw} has identical or better performance when n exceeds the number of variables, while benefiting from a better time complexity.

1 Introduction

The search for the true causal structure underlying some dataset is of paramount importance when the effect of interventions rather than predictions are to be returned (Pearl, 2000). Traditional statistical techniques cannot address problems where some parts of the data distribution process is changed. Causal modeling is a mean to address these nonstationary problems by computing the mechanism generating the data, usually as a directed acyclic graph (DAG), and by assessing the effect of some changes in that mechanism. But the task of detecting causation from observational data alone has long been a controversial issue: a strong statistical similarity between two variables cannot be equated with direct causation. It is also helpless for inferring the direction of causation. It is not before the pioneering work of Pearl and Verma (1991) and Spirtes *et al.* (1993) that causal discovery was formalized theoretically and linked with DAGs as graphical representation and mathematical object to reason on.

This causal search is in general impossible. When the distribution is restricted to be *DAG-isomorphic*, i.e., when all variable dependencies and independencies can be represented by a DAG (Wong *et al.*, 2002), part of it can be recovered from the data. Since the early 90s, a series of algo-

rithms have been proposed on that topic, of which PC[3] (Spirtes *et al.*, 1993) and IC (for "Inductive Causation") (Pearl and Verma, 1991) are two typical examples. Most of these algorithms have been designed for discrete variables, or impose restrictions on which variables may be continuous. The extension to exclusively continuous datasets is not straightforward, as the methods rely on very specific statistical tests. Some attempts have been made (e.g., Margaritis, 2005) but at the cost of increasing the time complexity by many degrees of magnitude, making generalizations of current approaches intractable for all but the smallest problems.

The current approach of choice when dealing with continuous variables is structural equation modeling (SEM), but it generally assumes that the structure is known in advance or defined by the user directly. Some work has been done to extend PC to the special case of recognizing the structure of a linear SEM (e.g., Scheines *et al.*, 1995), but speed is still an issue.

This technical report presents approaches to causal structure learning based on partial correlation and linear regression. We also use feature selection techniques to improve the robustness of the search when the sample size is small compared to the size of the network.

In section 2, we first review the principles of causal modeling and of causation detection with conditional independence tests, and go through the steps of PC, the reference algorithm. We then present our algorithms in section 3 and analyze their complexity. Experimental results are shown in section 4 and discussed in section 5. We finally conclude in section 6.

2 Background

We first briefly describe causal models in general, and then mention assumptions and properties that semantically link a DAG to its interpretation in terms of variable independence and causation.

2.1 Causal Models

The first step in a causal analysis is the definition of the causal structure traditionally represented as a DAG. (From now on, we make the *DAG-isomorphicity assumption* to ensure that the dataset to be analyzed can be represented by a DAG.) In this DAG, nodes represent the variables $\mathbf{V}$ of the dataset to be analyzed, possibly augmented by unobserved variables $\mathbf{U}$. Subsequent steps include choosing a causal model to put on top of the DAG describing the causal structure. Traditionally, this means either a *causal Bayesian network* or a *structural equations model* or *SEM* (Pearl, 2000):

- a Bayesian network (BN, a.k.a. belief network, Bayes net) is a tuple $\mathcal{B} = \langle \mathcal{G}, \mathbf{P} \rangle$, where $\mathcal{G} = \langle \mathbf{V}, \mathbf{E} \rangle$ is a DAG, and $\mathbf{P}$ a set of conditional probability distributions for each node in $\mathbf{V}$ given its graphical parents $\mathbf{Pa}(X)$;
- a SEM is a set of equations describing the value of each variable in $\mathbf{V}$ as a function f_X of its parents $\mathbf{pa}_X$ and a random disturbance term u_X:

$$x = f_X(\mathbf{pa}_X, u_X). \tag{1}$$

  When the graph obtained by drawing arcs from parents to children is acyclic, then the SEM is called *recursive*.

[3] "PC" stands for "Peter and Clark," after the two inventors of the method.

Usually, discrete-valued problems are represented as causal BNs, and continuous-valued problems with SEMs. This comes mainly from the inadequacy of representing the needed continuous conditional probability distributions in BNs, whereas the The final trained model can be used by tools such as the *do*-calculus (Pearl, 2000) to predict the effect of interventions or structural changes.

In either case, learning the structure of the graph representing the true causal structure is central to the approach. Detecting direct causation between a couple of variables, however, cannot be achieved with statistical method if only provided observational data, so that full identification of the causal graph is in general impossible. What we know though is that direct causation between two variables $X \to Y$ implies (unconditional) dependence, and that therefore (conditional) independence implies lack of direct causation. Conditional independence (CI) tests can thus help identify the causal structure by ruling out adjacencies in the causal graph[4]. CI is a ternary relation which is defined as follows.

Definition 2.1 (Conditional independence) *In a variable set* $\mathbf{V}$*, two random variables* $X, Y \in \mathbf{V}$ *are* conditionally independent *given* $\mathbf{Z} \subset \mathbf{V} \setminus \{X, Y\}$*, noted* $(X \perp\!\!\!\perp Y \mid \mathbf{Z})$*, if, for all* $x \in \mathcal{X}$*,* $y \in \mathcal{Y}$ *and* $\mathbf{z} \in \mathcal{Z}$*:* $p_{X|Y\mathbf{Z}}(X = x \mid Y = y,\ \mathbf{Z} = \mathbf{z}) = p_{X|\mathbf{Z}}(X = x \mid \mathbf{Z} = \mathbf{z})$.

If $(X \perp\!\!\!\perp Y \mid \mathbf{Z})$ holds, then we can write $P(X, Y \mid \mathbf{Z}) = P(X \mid \mathbf{Z})P(Y \mid \mathbf{Z})$ and find the classical definition of independence up to some conditioning set $\mathbf{Z}$.

We formalize the relation between direct causation (written $\to$) and conditional independence. Direct causation $X \to Y$ implies that X and Y are dependent in every context:

$$X \to Y \implies \big(\forall \mathbf{S} \subset \mathbf{V} \setminus \{X, Y\} : (X \not\perp\!\!\!\perp Y \mid \mathbf{S})\big). \tag{2}$$

The exact converse does not hold. If we make the *causal sufficiency assumption*, i.e., assume that no hidden common cause of two variables exist, we can write:

$$\big(\forall \mathbf{S} \subset \mathbf{V} \setminus \{X, Y\} : (X \not\perp\!\!\!\perp Y \mid \mathbf{S})\big) \implies X \to Y \text{ or } Y \to X. \tag{3}$$

Using (3), we can theoretically determine all adjacencies of the causal graph with CI tests, but we cannot orient the edges. But actually, there is a special causation pattern where CI test can reveal the direction of causation. It is known as a *V-structure*: two common causes X, Y, initially independent, become dependent when conditioned on a common effect Z, then known as a *collider*. Formally, we have:

$$X \to Z \leftarrow Y \implies \big(\exists \mathbf{S} \subset \mathbf{V} \setminus \{X, Y, Z\} : (X \perp\!\!\!\perp Y \mid \mathbf{S}) \text{ and } (X \not\perp\!\!\!\perp Y \mid \mathbf{S} \cup Z)\big). \tag{4}$$

The exact converse does not hold either. We have instead:

$$\begin{aligned} &\big(\exists \mathbf{S} \subset \mathbf{V} \setminus \{X, Y, Z\} : (X \perp\!\!\!\perp Y \mid \mathbf{S}) \text{ and } (X \not\perp\!\!\!\perp Y \mid \mathbf{S} \cup Z)\big) \\ \implies\ & X \leftrightarrow\cdot\to Z \leftarrow\cdot\leftrightarrow Y \text{ or } Z \text{ is an effect of some } W \text{ such that } X \leftrightarrow\cdot\to W \leftarrow\cdot\leftrightarrow Y. \end{aligned} \tag{5}$$

The notation $X \leftrightarrow\cdot\to Z$ means "any causal chain between X and Z pointing into Z that does not contain a V-structure." $X \to Z \leftarrow Y$ is a special case of $X \leftrightarrow\cdot\to Z \leftarrow\cdot\leftrightarrow Y$.

[4]There exist other, score-based approaches to structure learning, and mixed approaches using both CI tests and scores, which we do not discuss in this paper.

If we combine (3) and (5), we can find an equivalence relation defining a V-structure:

$$X \twoheadrightarrow Z \twoheadleftarrow Y \iff \Big(\big(\exists \mathbf{S} \subset \mathbf{V} \setminus \{X, Y, Z\} : (X \perp\!\!\!\perp Y \mid \mathbf{S}) \text{ and } (X \not\perp\!\!\!\perp Y \mid \mathbf{S} \cup Z)\big) \\ \text{and } \big(\forall \mathbf{S} \subset \mathbf{V} \setminus \{X, Z\} : (X \not\perp\!\!\!\perp Z \mid \mathbf{S})\big) \\ \text{and } \big(\forall \mathbf{S} \subset \mathbf{V} \setminus \{Y, Z\} : (Y \not\perp\!\!\!\perp Z \mid \mathbf{S})\big)\Big). \tag{6}$$

Actually, typical algorithms first establish the existence of a link between two variables by looking for a certificate equivalent to or implicating the premise of (3), and then look for orientation possibilities using (5).

Now that we know to what extend we can learn about causation with CI tests, we need to represent these results in DAG.

2.2 Graphs for Causation

The obvious property that we desire to have in the graphical representation is a one-to-one mapping between an arc between two nodes $X \to Y$ whenever we detect direct causation $X \twoheadrightarrow Y$. In order to interpret the graph easily, it is also desirable to find a graphical criterion in bijection with the conditional independence relation when the arcs represent causation. This has been investigated (Pearl, 1988) and is known as *d-separation*.

Definition 2.2 (*d*-separation) *In an DAG $\mathcal{G}$, two nodes X, Y are* d-separated *by $\mathbf{Z} \subset \mathbf{V} \setminus \{X, Y\}$, written $(X \perp\!\!\!\perp_{\mathcal{G}} Y \mid \mathbf{Z})$, if every path from X to Y is blocked by $\mathbf{Z}$. A path is* blocked *if at least one diverging or serially connected node in in $\mathbf{Z}$ or if at least one converging node and all its descendants are not in $\mathbf{Z}$. If X and Y are not* d-*separated by $\mathbf{Z}$, they are* d-*connected: $(X \leftrightarrow Y \mid \mathbf{Z})$.*

Our goal in structure learning is then to find a graph which is a *perfect map* of the probability distribution sampled by the dataset.

Definition 2.3 (Perfect map) *A DAG $\mathcal{G}$ is a* directed perfect map *of a joint probability distribution $p(\mathbf{V})$ if there is bijection between* d-*separation in $\mathcal{G}$ and conditional independence in p:*

$$\forall X, Y \in \mathbf{V}, \forall \mathbf{Z} \subset \mathbf{V} \setminus \{X, Y\} : \big((X \perp\!\!\!\perp_{\mathcal{G}} Y \mid \mathbf{Z}) \iff (X \perp\!\!\!\perp Y \mid \mathbf{Z}) \big). \tag{7}$$

If we have a perfect map of our data, then we can answer any query about conditional independence. We use the perfect map property to prove that structure learning algorithms are correct. This property is defined in terms of conditional independence and not causation, and as we note in the previous section, all causal relations cannot be identified by CI statements in general. This is the problem known as *causal underidentification*: for the structure learning task given observational data, a correct graph is specified by its adjacencies and its V-structures only. Not all edges can be causally oriented. Graphs returned by structure learning algorithms are often partially directed acyclic graphs (PDAGs) and represent *observationally equivalent classes* of causal graphs (Pearl, 2000, p. 19). This means that for a given joint probability distribution $p(\mathbf{V})$, the set $\mathcal{ID}(p)$ of all conditional independence statements that hold in p yields does not yield a unique perfect map in general.

Formally, if we combine (7), (3) and (5), we find, for a perfect causal map:

$$X, Y \text{ adjacent in } \mathcal{G} \iff X \twoheadrightarrow Y \text{ or } Y \twoheadrightarrow X \tag{8}$$

$$X \to Z \leftarrow Y \iff X \twoheadrightarrow Z \twoheadleftarrow Y. \tag{9}$$

It is sometimes possible to orient further arcs in a graph by looking at already oriented arcs and propagating constraints such that acyclicity, or the fact that any further orientation may not create additional V-structure other than those already detected. The graph after this constraint propagation step is called *completed PDAG* or *CPDAG*.

2.3 PC Algorithm

We now turn to the typical example of a structure learning algorithm. The PC algorithm is These algorithms typically follow the three steps detailed below, following the high-level description of the IC algorithm (Pearl, 2000):

The PC algorithm uses CI tests to identify the graph up to observational equivalence. We present its high-level description textual description, also known as the IC (for "Inductive Causation") algorithm (Pearl and Verma, 1991). Note how Steps 1 and 2 match the implications in (3) and (5).

1. For each variable pair (X, Y) in the set of variables $\mathbf{V}$, look for a set $\mathbf{S}_{XY}$ such that X and Y are conditionally independent given $\mathbf{S}_{XY}$: $(X \perp\!\!\!\perp Y \mid \mathbf{S}_{XY})$; add an edge between X and Y if no such set can be found;
2. For each pair (X, Y) with a common neighbor Z in the identified graph skeleton, turn the triple into a V-structure $X \rightarrow Z \leftarrow Y$ if $Z \in \mathbf{S}_{XY}$;
3. Propagate the arrow orientation to preserve acyclicity without introducing new V-structures.

Step 1 is can be prohibitive: it is a subset search, which is exponential in time in the worst case of a fully connected graph. PC accepts an optional *maximum fan-in* parameter $d^\star$ which prevents it from testing conditioning subsets whose cardinality is greater than $d^\star$, thus limiting the complexity to $\mathcal{O}(2^{d^\star})$ instead of $\mathcal{O}(2^d)$.

In practice, however, with sparse graphs, most of the edges can be ruled out with rather small conditioning sets, such that the approach is still realistic. There are local patterns, however, which considerably slow PC down, as we shall see in section 4.

Pseudocode for the PC algorithm is listed in Algorithm 1. We created a separate procedure for Step 3 in Algorithm 2 as it will be reused later. The notation $\mathbf{Bd}(X)$ stands for the *boundary* of the node X; i.e., the set of nodes having an edge in common with X. The set of rules implementing constraint propagation have been shown to yield the maximally oriented PDAG, provided all V-structures have been identified prior to Step 3 (Meek, 1995).

2.4 Partial Correlation as Conditional Independence Measure

While the PC-like approach works well with discrete variables, continuous variables are more computationally expensive to deal with, specifically because there is no convincing general straightforward statistical test of CI for continuous variables. But we can find one if we look at the special case of *linear recursive SEMs*, where each functional equation is of the form

$$x = \mathbf{w}_X^T \mathbf{pa}_X + u_X, \tag{10}$$

which is a special case of the general SEM described by (1). Imposing a Gaussian distribution on the disturbance terms u_i yields a multivariate Gaussian distribution over $\mathbf{V}$. Then, *partial correlation* is a valid CI measure.

Algorithm 1 The PC algorithm

procedure PCSTRUCTURELEARNING
Input: $D : n \times d$ dataset with n d-dimensional data points
$d^\star$: (optional) maximum fan-in parameter; default is $d-2$
Output: $\mathcal{G}$: maximally oriented partially directed acyclic graph

/* Initialization */
$\mathcal{G} \leftarrow$ fully connected, undirected graph with d nodes
$i \leftarrow 0$
/* Step 1: Unnecessary arc deletion */
while $i \leq$ maximum number of edges for any node **and** $i \leq d^\star$ **do**
for each adjacent unordered pair X, Y such that $|\mathbf{Bd}(X)| > i$ **do**
if $\exists$ set $\mathbf{S} \subset \mathbf{Bd}(X)$ of size i such that $(X \perp\!\!\!\perp Y \mid \mathbf{S})$ **then**
remove link $X - Y$ from $\mathcal{G}$
$\mathbf{S}_{XY}, \mathbf{S}_{YX} \leftarrow \mathbf{S}$
end if
end for
$i \leftarrow i + 1$
end while
/* Step 2: V-structure detection */
for each X, Y, Z such that $X - Z - Y$ **do**
if $Z \notin \mathbf{S}_{XY}$ **then** orient as $X \rightarrow Z \leftarrow Y$
end for
/* Step 3: Constraint propagation */
$\mathcal{G} \leftarrow$ COMPLETEPDAG$(\mathcal{G})$
return $\mathcal{G}$
end procedure

Definition 2.4 (Partial correlation) *In a variable set* $\mathbf{V}$*, the* partial correlation *between two random variables* $X, Y \in \mathbf{V}$ *given* $\mathbf{Z} \subset \mathbf{V} \setminus \{X, Y\}$*, noted* $\rho_{XY \cdot \mathbf{Z}}$*, is the correlation of the residuals* R_X *and* R_Y *resulting from the least-squares linear regression of* X *on* $\mathbf{Z}$ *and of* Y *on* $\mathbf{Z}$*, respectively.*

In the multivariate Gaussian case, $\rho_{XY \cdot \mathbf{Z}} = 0$ if and only if $(X \perp\!\!\!\perp Y \mid \mathbf{Z})$ holds (Baba *et al.*, 2004). Partial correlation can be computed efficiently without having to solve the regression problem by inverting the correlation matrix $\mathbf{R}$ of the union $\mathbf{Z} \cup \{X, Y\}$. With $\mathbf{R}^{-1} = (r^{ij})$, we have: $\rho_{X_i X_j \cdot \mathbf{V} \setminus \{X_i, X_j\}} = -r^{ij} / \sqrt{r^{ii} r^{jj}}$ (Raveh, 1985). This is in particular convenient if several tests use the same set of variables formed by the union of the three arguments. In this case, we can compute all partial correlations with a single matrix inversion. This is an approach we use in our algorithm.

3 Total Conditioning for Causal Discovery

We first describe the TC algorithm based on the result of equivalence of zero partial correlation and CI for linear SEMs. We then present an extension, TC_{bw}, which consists of adding a feature selection step to make it more robust when the number of samples n gets low and approaches the number of variables d.

Algorithm 2 Turn a PDAG into its corresponding CPDAG

procedure COMPLETEPDAG
Input: $\mathcal{G}$: partially directed acyclic graph
Output: $\mathcal{G}$: maximally oriented partially directed acyclic graph

/* Constraint propagation */
while $\mathcal{G}$ is changed by some rule **do** /* fixed-point iteration */
for each X, Y, Z such that $X \to Y - Z$ **do**
orient as $X \to Y \to Z$ /* no new V-structure */
end for
for each X, Y such that $X - Y$ and $\exists$ directed path from X to Y **do**
orient as $X \to Y$ /* preserve acyclicity */
end for
for each X, Y s.t. $X - Y$ and $\exists$ nonadjacent Z, W s.t. $X - Z \to Y$ and $X - W \to Y$ **do**
orient as $X \to Y$ /* three-fork V with married parents */
end for
end while
end procedure

3.1 TC Algorithm

Whereas PC removes edges from a full graph as CI is found, our Total Conditioning (TC) method starts with an empty graph and adds edges between two nodes when conditioning on all the others does not reveal independency. In addition to the textual description, we list pseudocode in Algorithm 3.

1. For each pair (X, Y), add an edge $X - Y$ if the partial correlation $\rho_{XY \cdot \mathbf{V} \setminus \{X,Y\}}$ does not vanish. We obtain the *moral graph* of the original DAG $\mathcal{G}_0$, i.e., an undirected copy of $\mathcal{G}_0$ where all parents of the colliders are pairwise linked;
2. Remove spurious links between parents of colliders introduced in Step 1 and identify V-structures;
3. Propagate constraints to obtain maximally oriented graph (completed PDAG).

In terms of Gaussian Markov random fields (a special case of undirected graphical models), Step 1 constructs the correct graph by adding edges where the total partial correlation is significantly different from zero (e.g., Talih, 2003). Step 3 is common to several algorithms constructing the network structure under CI constraints and is the same as in the IC/PC algorithms. Step 2 is a local search looking for orientation possibilities; to explain it, we need the following definition.

Definition 3.1 (Collider Set Property) *In an undirected graph $\mathcal{G} = (\mathbf{V}, \mathbf{E})$, let $\mathbf{Tri}(X - Y)$ (with $X, Y \in \mathbf{V}$ and $(X, Y) \in \mathbf{E}$) be the set of vertices forming a triangle with X and Y:*

$$\mathbf{Tri}(X - Y) = \{Z \in \mathbf{V} \mid \{(X, Z), (Y, Z)\} \subset \mathbf{E}\}. \tag{11}$$

Suppose that $\mathcal{G}$ is the moral graph of the DAG representing the causal structure of some DAG-isomorphic dataset. A set of vertices $\mathbf{Z} \subset \mathbf{Tri}(X - Y)$ then has the Collider Set *property for the*

pair (X, Y) *if it is the largest set that fulfills*

$$\exists \mathbf{S}_{XY} \subset \mathbf{V} \setminus \{X, Y\} \setminus \mathbf{Z} : (X \perp\!\!\!\perp Y \mid \mathbf{S}_{XY}) \tag{12}$$

$$\textit{and} \quad \forall Z_i \in \mathbf{Z} : (X \not\perp\!\!\!\perp Y \mid \mathbf{S}_{XY} \cup Z_i)\,. \tag{13}$$

The set $\mathbf{S}_{XY}$ *is then a* d*-separating set for* X, Y*.*

Algorithm 3 The Total Conditioning algorithm

```
1: procedure TCStructureLearning
   Input:  D : n × d dataset with n d-dimensional data points
   Output: G : maximally oriented partially directed acyclic graph
     /* Step 1: Skeleton construction */
2:   G ← empty graph with d nodes
3:   for each unordered pair X, Y do
4:     if ρ_{XY·V\{X,Y}} does not vanish then add link X − Y to G
5:   end for
     /* Step 2: Spurious arc removal & V-structure detection */
6:   for each edge X − Y part of a fully connected triangle do
7:     S_XY ← ColliderSetSearch(G, X, Y)
8:     if S_XY ≠ null then
9:       remove link X − Y from G
10:      for each Z ∈ (Tri(X − Y) \ S_XY) do orient edges as X → Z_i ← Y
11:    end if
12:  end for
     /* Step 3: Constraint propagation */
13:  G ← CompletePDAG(G)
14:  return G
15: end procedure
```

Step 1 of the algorithm builds the correct structure up to moral graph equivalence; it will actually build the correct undirected links and marry all parents. This means that every original V-structure will be turned into a triangle.

Step 2 looks at each edge that is part of some triangle and determines if it is spurious due to a V-structure effect. This is exactly the case when two variables X, Y in a triangle X, Y, Z can be made conditionally independent by a set that does not contain Z. A search is then performed for each of those edges to determine a set $\mathbf{Z} \subset \mathbf{Tri}(X - Y)$ that has the Collider Set property, using a small search space for $\mathbf{S}_{XY}$ and $\mathbf{Z}$ as allowed by the result of Step 1. If this search is successful, the edge $X - Y$ is removed and the detected V-structures properly oriented for each collider. Practically, the search for $\mathbf{S}_{XY}$ can be restricted to a subset of the union of the Markov blankets for X and Y, and the search for $\mathbf{Z}$ is restricted by definition to $\mathbf{Tri}(X - Y)$, which make both tasks tractable, unless the graph has a high connectedness.

Practically, we look for this set by using tests of zero partial correlation with, as conditioning sets, all nodes that are not on a path of length 2 (i.e., on the only path that could be open by

conditioning on all its nodes), to which we successively add the nodes lying on the paths of length 2. The search terminates when CI is found or when all possibilities have been visited unsuccessfully. The nodes that end up not being in the d-separating set are then identified as colliders and the corresponding edges oriented accordingly. Pseudocode for this search is listed in Algorithm 4. This specific procedure actually looks for a d-separating set, and the collider set is then inferred by removing the d-separating set from the triangle nodes $\mathbf{Tri}(X - Y)$.

Note that returning an empty set is different from returning **null**. An empty set means that X and Y are unconditionally dependent, and therefore, that all nodes in $\mathbf{Tri}(X - Y)$ are colliders and that the link between X and Y was spurious. Returning **null**, on the contrary, indicates that no d-separating set could be found and this that there actually is a link between X and Y.

Algorithm 4 Search for a d-separating set determining the Collider Set

1: **procedure** COLLIDERSETSEARCH
   Input: $\mathcal{G}$: graph possibly containing spurious edges
   X, Y : nodes in $\mathcal{G}$ (possibly spuriously) linked
   Output: $\mathbf{Z}$: d-separating set for X and Y, or **null** if search failed
2: $\mathbf{B} \leftarrow \big(\mathbf{Bd}(X) \cup \mathbf{Bd}(Y)\big) \setminus \mathbf{Tri}(X - Y)$ /* Base conditioning set */
3: **for each** $\mathbf{S} \subset \mathbf{Tri}(X - Y)$ **do** /* Subset search */
4: $\mathbf{Z} \leftarrow \mathbf{B} \cup \mathbf{S}$
5: **if** $\rho_{XY \cdot \mathbf{Z}}$ vanishes **then return** $\mathbf{Z}$
6: $\mathbf{D} \leftarrow \mathbf{B} \cap \{\text{possible descendants of } W \mid W \in \big(\mathbf{Tri}(X - Y) \setminus \mathbf{S}\big)\}$
7: $\mathbf{B}' \leftarrow \mathbf{B} \setminus \mathbf{D}$
8: **for each** $\mathbf{S}' \subset \mathbf{D}$ **do** /* Descendant of collider may be opening a path */
9: $\mathbf{Z} \leftarrow \mathbf{B}' \cup \mathbf{S}' \cup \mathbf{S}$
10: **if** $\rho_{XY \cdot \mathbf{Z}}$ vanishes **then return** $\mathbf{Z}$
11: **end for**
12: **end for**
13: **return null**
14: **end procedure**

The rationale behind Algorithm 4 is that we look for d-separation given a certain subset, and every node not in this subset but part of a triangle with X and Y must then be a collider. Two caveats have to be observed, however. First, there might be other active, d-connecting paths between X and Y that are not going through any node of $\mathbf{Tri}(X - Y)$. Those nodes must be blocked by appropriate conditioning on the boundary of X and Y as determined by the base conditioning set on line 2. Second, this base conditioning set must be checked not to include any descendant of possible colliders. If it does, it opens a d-connecting path according to Definition 2.2. This check is performed in lines 6 to 11. At line 6, we build a set $\mathbf{D}$ that includes all possible descendants of currently conjectured colliders that intersect our base conditioning set $\mathbf{B}$. The following loop makes sure none of them was opening a path between X and Y.

Step 1 of the TC algorithm has a complexity of $\mathcal{O}(d^3)$, which comes from the matrix inversion needed to compute the partial correlations. Step 2 has a complexity of $\mathcal{O}(d^2 2^\alpha)$, where $\alpha = \max_{X,Y} |\mathbf{Tri}(X - Y)| - 1$. In the worst case of a fully connected graph, where $\mathbf{Tri}(X - Y) =$

$\mathbf{V} \setminus \{X, Y\}$, it is exponential in the number of variables. Step 3 is $\mathcal{O}(d^3)$. The overall complexity is then $\mathcal{O}(d^3 + d^2 2^{\alpha})$, depending on the value of α as determined by the structure of the graph to be recovered.

After removal of the spurious links and the constraint propagation step, the returned graph is the maximally oriented PDAG of the equivalence class of the generating DAG $\mathcal{G}_0$. We prove this result in the appendix.

3.2 Significance Tests

A delicate point in TC is the statistical test deciding if a partial correlation is significant. In practice, we replaced the more traditional Fisher approximate z-transform of the sample correlation by t-tests on the weights of the linear regression problems used in the definition of partial correlation. Their distributions are known to be Gaussian with zero mean under the null hypothesis (e.g., Judge *et al.*, 1988, p. 243). They can also be computed efficiently from the inverse correlation matrix $\mathbf{R}^{-1}$ (Raveh, 1985). The choice of the Type I error α needs investigating as it significantly influences the result of the algorithm.

In a network of d nodes, Step 1 performs $d(d-1)/2$ tests to determine the undirected skeleton. We will falsely reject the null hypothesis $\rho = 0$ about $m \cdot \alpha$ times on average, where $m < d(d-1)/2$ is the difference in the number of edges between the original DAG $\mathcal{G}_0$ and the complete graph. We will thus add on average $m \cdot \alpha$ wrong edges. We can set the significance level for the individual tests to be inversely proportional to $d(d-1)/2$ to avoid this problem (assuming a large m and thus rather sparse graphs), and check that it does not affect the Type II error rate too much, which we do now.

Still assuming Gaussian disturbance terms on a linear SEM, the structure learning problem can be viewed as d multiple regression problems. Call $\hat{b}_{ik}$ the maximum likelihood estimator of the regression weight b_{ik} of predictor k when i is the dependent variable. Then it can be shown (Judge *et al.*, 1988) that

$$\frac{\hat{b}_{ik} - b_{ik}}{\hat{\sigma}_{ik}} \sim t_{(n-(d-1))}, \tag{14}$$

where $\hat{\sigma}_{ik}$ is the standard error of predictor k for variable i; i.e., that it follows a t distribution with a number of degrees of freedom df = number of samples − number of predictors $= n - (d-1)$. If we call $\Psi(\cdot)$ the cumulative distribution function of a t distribution with $(n-(d-1))$ degrees of freedom, we can write the Type II error rate β for each regression weight:

$$\beta_{ik} = \Psi(\Psi^{-1}(1 - \alpha/2) - |b_{ik}|/\hat{\sigma}_{ik}). \tag{15}$$

The values for $\hat{\sigma}_{ik}$ can be computed from the inverse correlation matrix $\mathbf{R}^{-1}$ and thus depend on the particular dataset being analyzed, but the true b_{ik} are unknown. What we could do in theory to optimize α is to minimize the average number of extraneous (T_e) and missing (T_m) links:

$$T = T_e + T_m = m \cdot \alpha + \sum_{(i,k) \in I} \beta_{ik}, \tag{16}$$

where m is the number of edges missing in the original DAG compared to a full graph, and I is the set of arcs in the original DAG, so that $m + |I| = d(d-1)/2$. As m, I and b_{ik} are unknown, we can only find an upper bound for the number of missed links T_m, provided (i) we can estimate the

graph sparseness to approximate m; (ii) we assume $|b_{ik}| \geq \delta$; and (iii) we choose $I^\star$ such that it maximizes the sum in (17), with $|I^\star| = d(d-1)/2 - m$. Then we have:

$$T_m \leq \sum_{(i,k)\in I^\star} \Psi(\Psi^{-1}(1-\alpha/2) - \delta/\hat{\sigma}_{ik}). \tag{17}$$

Although this bound was found too loose for practical use, we can the Type I and Type II errors rate as a function of α for artificial problems whose sparseness and regression weights are known. This is shown in Figure 1 for a specific instance of an Alarm dataset with two different sample sizes (see section 4). We did not use this information to tune α in the experiments, as it cannot be obtained without prior knowledge, but the curves showed that an α inversely proportional to $d(d-1)/2$ has the same order of magnitude as the optimal α on the datasets we analyzed.

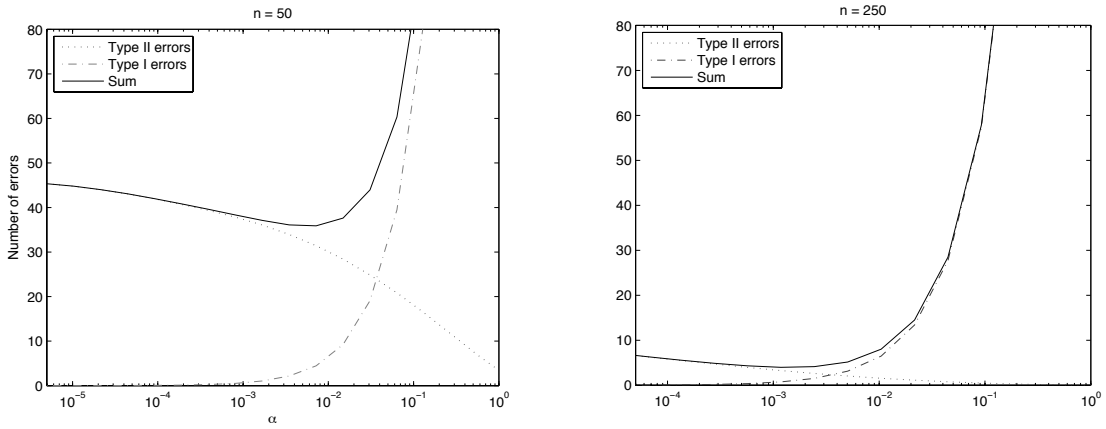

FIG. 1: Expected Type I and II errors as a function of α

3.3 TC_{bw} Algorithm

Despite correctness of TC, with a low number of samples n it fails to have enough evidence for rejecting the null hypothesis of zero partial correlation (or zero regression weight), and thus misses links (see detailed results in section 4), even for a high α. We now try to address this particular issue by successively eliminating the most insignificant predictors and reevaluating the remaining ones. This is actually a backwards stepwise regression method. Steps 2 and 3 remain identical to those of TC, and Step 1 is changed as follows:

1. For each variable X, solve the multiple regression problem considering the independent variables $\mathbf{V} \setminus X$. Sort predictors according to significance. Loop:
   (a) Remove the p most insignificant from the list of independent variables and solve the reduced multiple regression problem.
   (b) Loop until every predictor is significant or the set of predictors is empty.
   Add an edge between X and every significant predictor.

Pseudocode for the full algorithm is listed in Algorithm 5.

Algorithm 5 The Total Conditioning with Feature Selection algorithm

procedure TCBWSTRUCTURELEARNING
Input: D : $n \times d$ dataset with n d-dimensional data points
Output: $\mathcal{G}$: maximally oriented partially directed acyclic graph

/* Step 1: Skeleton construction */
$\mathcal{G} \leftarrow$ empty graph with d nodes
for each variable X **do**
  $\mathbf{P} \leftarrow \mathbf{V} \setminus X$ /* all predictors */
  $\mathbf{S} \leftarrow \emptyset$ /* significant predictors */
  while $\mathbf{P} \neq \emptyset$ **and** $\mathbf{P} \neq \mathbf{S}$ **do**
    $\mathbf{b} \leftarrow$ weights of $\mathbf{P}$ in the problem of regressing X on $\mathbf{P}$
    $\mathbf{S} \leftarrow \mathbf{S} \cup \{$predictors whose b weight is significant$\}$
    $\mathbf{P} \leftarrow \mathbf{P} \setminus \{$the p less significant predictors$\}$
  end while
  for each $S_i \in \mathbf{S}$ **do** add link $X - S$
end for
/* Step 2: Spurious arc removal & V-structure detection */
for each edge $X - Y$ part of a fully connected triangle **do**
  $\mathbf{S}_{XY} \leftarrow$ COLLIDERSETSEARCH$(\mathcal{G}, X, Y)$
  if $\mathbf{S}_{XY} \neq$ **null then**
    remove link $X - Y$ from $\mathcal{G}$
    for each $Z \in \left(\mathbf{Tri}(X - Y) \setminus \mathbf{S}_{XY}\right)$ **do** orient edges as $X \rightarrow Z \leftarrow Y$
  end if
end for
/* Step 3: Constraint propagation */
$\mathcal{G} \leftarrow$ COMPLETEPDAG$(\mathcal{G})$
return $\mathcal{G}$
end procedure

Intuitively, the problem to solve is that the regression weights cannot be high enough for significance with small sample sizes. By removing the most insignificant predictors and thus the most likely to be actually zero, we scale down the regression problem and augment the power of the tests. How many insignificant to remove can be discussed; in our implementation, we compared $p = 1$ to $p = (\#$ of predictors$)/2$ and found that the latter yielded as good results with an important speed gain.

Solving a standard multiple regression problem with d predictors traditionally has complexity $\mathcal{O}(d^3)$. Naïvely solving $d - 1$ regression problems d times in the case $p = 1$ would have a complexity of $\mathcal{O}(d^5)$. But we can avoid reinverting matrices in the inner loop of Step 1 thanks to the following result.

Let $\mathbf{\Sigma} = \mathbf{X}^T\mathbf{X}$ be n times the correlation matrix $\mathbf{R}$, where $\mathbf{X}$ is the $n \times d$ matrix representing a dataset where all variables have zero mean and unit standard deviation. Then we can use $\mathbf{\Sigma}^{-1}$ to linearly find the weights of the regression problems, their standard error, and the mean squared error of the regression. Suppose we find that variable X_1 is the weakest predictor, and want to reevaluate the weights of the other predictors in Step 1 (a) of TC_{bw}. Let $\mathbf{X}_{\setminus i}$ be the dataset where

variable X_i has been removed. Then we need the matrix $\mathbf{\Omega}^{-1}$ to solve the new problem, where $\mathbf{\Omega} = \mathbf{X}_{\setminus 1}^T \mathbf{X}_{\setminus 1}$. We have:

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_{11} & \mathbf{c}^T \\ \mathbf{c} & \mathbf{\Omega} \end{bmatrix} \quad \Longrightarrow \quad \mathbf{\Sigma}^{-1} = \begin{bmatrix} \frac{1}{\sigma_{11} - \mathbf{c}^T \mathbf{\Omega}^{-1} \mathbf{c}} & -\frac{\mathbf{c}^T \mathbf{\Omega}^{-1}}{\sigma_{11} - \mathbf{c}^T \mathbf{\Omega}^{-1} \mathbf{c}} \\ -\frac{\mathbf{\Omega}^{-1} \mathbf{c}}{\sigma_{11} - \mathbf{c}^T \mathbf{\Omega}^{-1} \mathbf{c}} & \mathbf{\Omega}^{-1} + \frac{\mathbf{\Omega}^{-1} \mathbf{c} \mathbf{c}^T \mathbf{\Omega}^{-1}}{\sigma_{11} - \mathbf{c}^T \mathbf{\Omega}^{-1} \mathbf{c}} \end{bmatrix}. \tag{18}$$

Let $\sigma^{ij} = (\mathbf{\Sigma}^{-1})_{ij}$ and $\mathbf{b} = \mathbf{\Omega}^{-1}\mathbf{c}$. Then $\mathbf{b}$ are the weights of the regression of X_1 on $X_2, \cdots, X_d$ and can be computed without knowing $\mathbf{\Omega}^{-1}$ (Raveh, 1985):

$$\mathbf{b} = (b_j) = (-\sigma^{1j}/\sigma^{11}). \tag{19}$$

We can thus compute $\mathbf{\Omega}^{-1}$ given $\mathbf{\Sigma}^{-1}$ with complexity $\mathcal{O}(d^2)$ as follows:

$$\mathbf{\Omega}^{-1} = (\mathbf{\Sigma}^{-1})_{\setminus 1} - \sigma^{11} \mathbf{b}\mathbf{b}^T, \tag{20}$$

with $(\mathbf{\Sigma}^{-1})_{\setminus 1}$ being the matrix $\mathbf{\Sigma}^{-1}$ where the first row and column have been removed. In the case $p = 1$, this elimination of row and column of the inverse matrix is repeated at most $d - 1$ for each variable, yielding a complexity of $\mathcal{O}(d^4)$. This overall complexity of TC_{bw} is then $\mathcal{O}(d^4 + d^2 2^\alpha)$.

4 Experimental Results

The performance of the TC and TC_{bw} algorithms was evaluated against the PC algorithm (Spirtes *et al.*, 1993) where CI tests were replaced by zero partial correlation tests. We were unable to compare it to newer algorithms like SCA (Friedman *et al.*, 2000) or MMHC (Tsamardinos *et al.*, 2006) because generalizing them to handle continuous variables require techniques that are too computationally expensive, notably because of hard to generalize score-based subroutines. We used the following networks (from the Bayes net repository):

- Alarm (Beinlich *et al.*, 1989), 37 nodes, 46 arcs, 4 undirected in the PDAG of the equivalence class. It was originally designed to help interpret monitoring data to alert anesthesiologists to various situations in the operating room. It is depicted in Figure 2.
- Hailfinder (Abramson *et al.*, 1996), 56 nodes, 66 arcs, 17 undirected in its PDAG. It is a normative system that forecasts severe summer hail in northeastern Colorado. See Figure 3.
- A subset of Diabetes (Andreassen *et al.*, 1991) with 104 nodes, 149 arcs, 8 undirected in its PDAG, which was designed as a preliminary model for insulin dose adjustment. This subset is made of 6 repeating patterns (there are 24 in the original network) of 17 nodes, plus 2 external nodes linked to every pattern. The first two of these patterns are shown in Figure 4.

The graphs were used as a structure for a linear SEM. The parentless variables were sampled as Gaussians with zero mean and unit standard deviation; the other variables were defined as a linear combination of their parents with coefficient randomly distributed uniformly between 0.2 and 1, similarly to what was done in Scheines *et al.* (1995). The disturbance terms were also normally distributed. We used the implementation of PC proposed by (Leray and François, 2004) in the BNT Structure Learning Matlab package, where we set the statistical significance of the tests to $\alpha = 0.05$. In TC and TC_{bw}, we set $\alpha = 2/(d(d-1))$. The implementation of TC and TC_{bw} was also done in Matlab; all experiments were run on a 2 GHz machine.

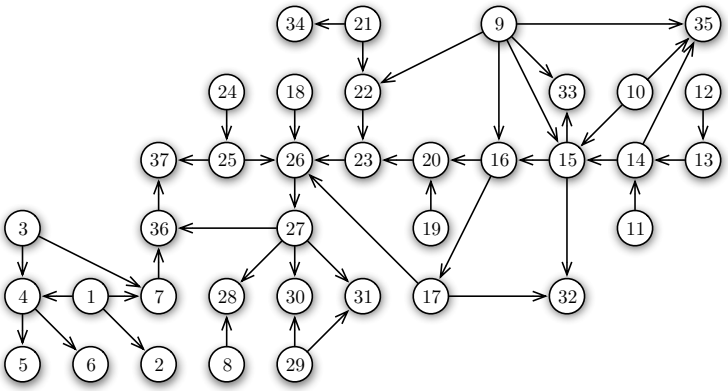

FIG. 2: The Alarm network

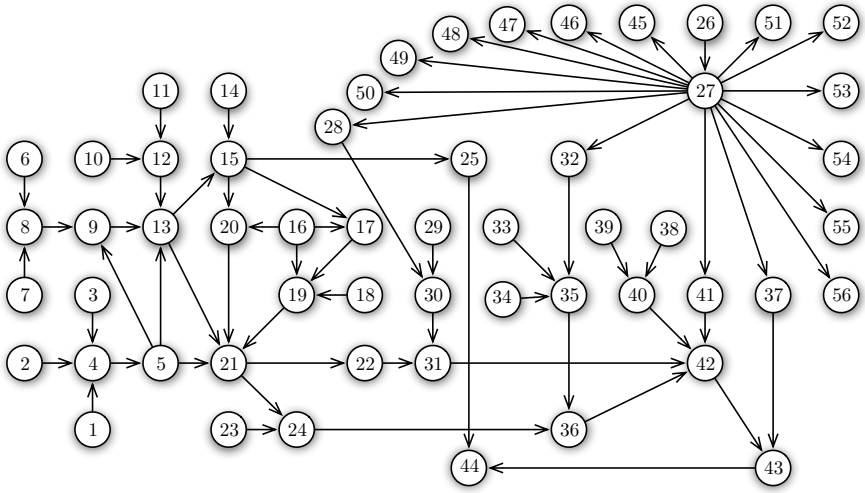

FIG. 3: The Hailfinder network

Figure 5 (a) shows the training errors against the number of samples for Alarm. For each sample size, 9 datasets were drawn from the model; the error bars picture the standard deviation over these 9 runs. Starting at about 150 samples, TC outperforms PC. TC_{bw} beats both TC and PC, and the converging curves of TC and TC_{bw} show that the stepwise regression becomes unnecessary near 300 samples. On average, TC was about 20 times faster than the implementation of PC we used, although the factor tended to decrease with larger sample sizes; see Figure 5 (b). TC_{bw} was naturally slower than TC, although marginally compared to the speed difference with PC.

The results for Hailfinder are shown in Figure 6. The results for PC are sparser than for TC, because of its long run times. In order to speed it up, we set the maximum node fan-in parameter to 6, so that PC would not attempt to conduct CI tests with conditioning sets larger than 6. For large datasets, we could run PC only once, so that we have little information on the variance of its results for this network, but even if we average the five last PC results for between 200 and 5000 samples, TC does better for each of its runs on this range. TC_{bw} performs better everywhere except under about 70 samples, and joins TC at around 1000 samples. Still, PC still beats TC on smaller

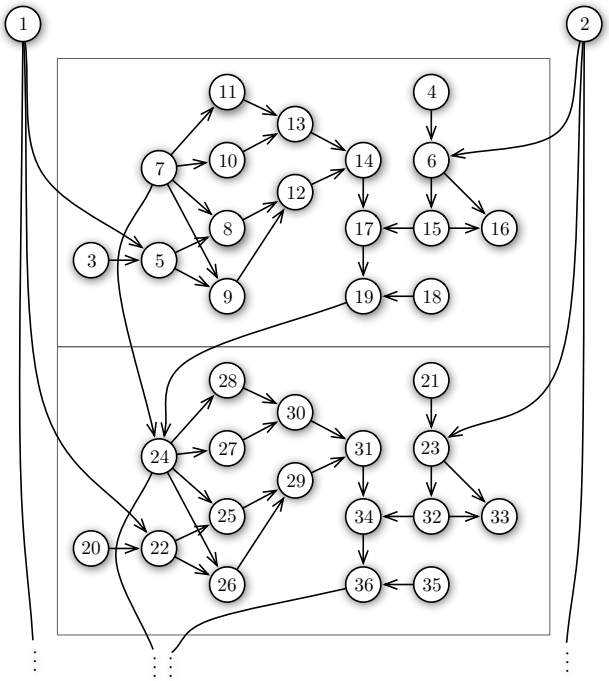

FIG. 4: Two of the six patterns of the Diabetes network

sample sizes. We also see on Figure 6 (b) how the fan-in parameter imposed an upper bound on the run times of PC.

Figure 7 shows errors and run times for Diabetes. On this network, PC is beaten from 2500 samples on Again, PC does better at first, and starting at 500 samples, it is outperformed in accuracy. The difference of the number of errors stabilizes around 5 or 6. Both TC and TC_{bw} achieve significantly shorter run times than PC. Here again, the error curves of TC and TC_{bw} join at about 1000 samples.

Finally, Figure 8 shows the results of an experiment where we took the first d nodes of Diabetes for a fixed ratio $n/d = 20$ in order to show the response of the algorithms to an increasing number of variables in networks of similar structure. Results show that the three algorithms make a similar number of mistakes (plus minus one) generally, although PC seems to be outperformed consistently for growing d on this particular instance. The run times of PC are still significantly higher. But more interesting is in Figure 9 (a) the number of statistical tests performed by the algorithms for the same experiment, and in Figure 9 (b) the scaled version, where the number of tests have been divided by the maximum number of tests for the largest network. We read off (b) that even if scaled down by some factor, the number of tests grows faster for PC than for TC and TC_{bw}, as the derivative of the curve for PC is larger as d increases. We find that both our algorithms scale better than PC on this instance.

5 Discussion

Both TC and TC_{bw} consistently beat PC when the sample size gets larger in a small fraction of its run time. In particular, PC is slowed down by nodes with a high degree, whereas TC handles them

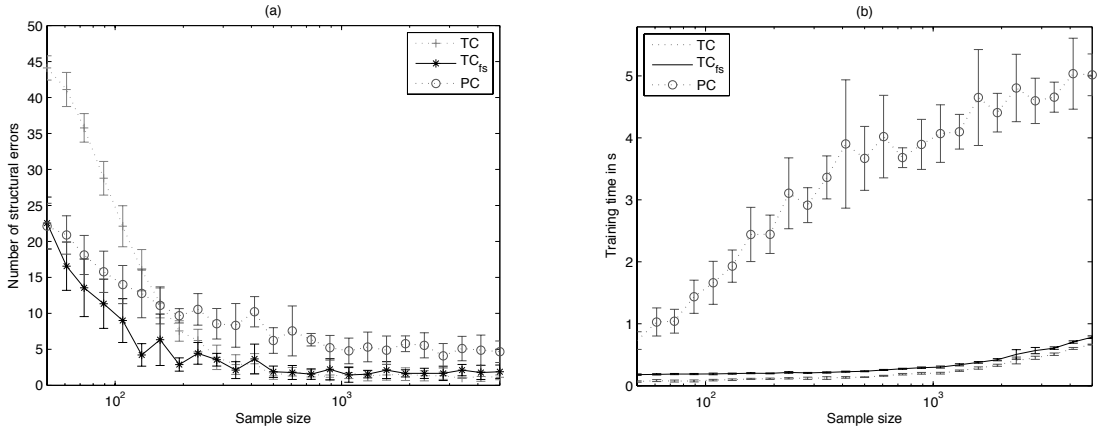

FIG. 5: Alarm: (a) structural errors and (b) run times as a function of sample size

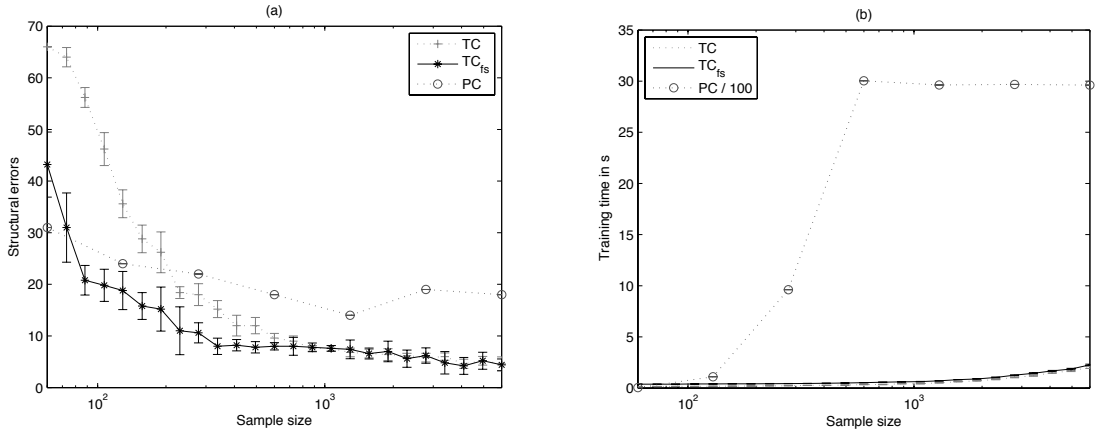

FIG. 6: Hailfinder: (a) structural errors and (b) run times as a function of sample size

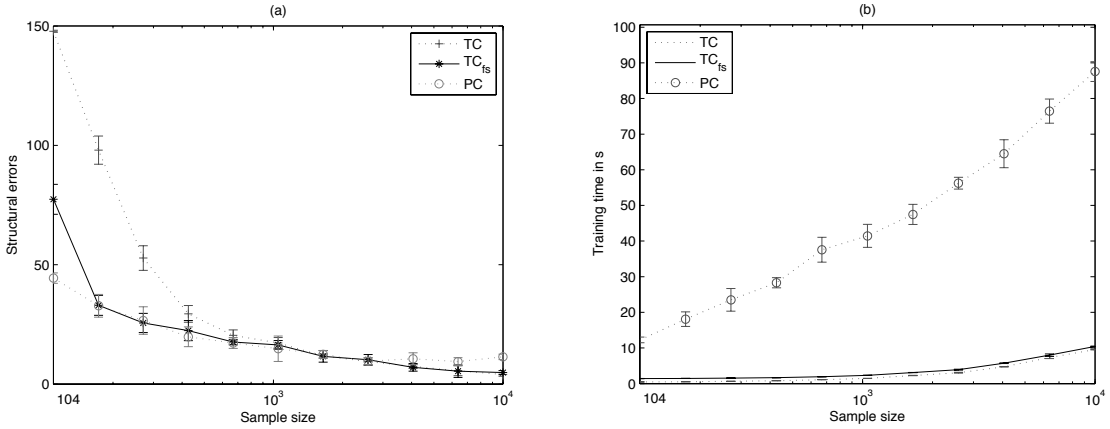

FIG. 7: Diabetes: (a) structural errors and (b) run times as a function of sample size

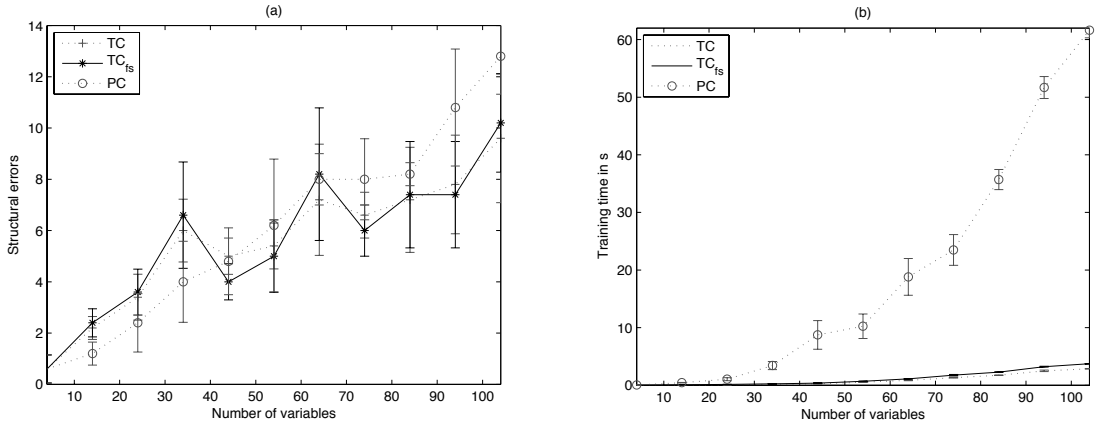

FIG. 8: Diabetes: (a) structural errors and (b) run times as a function of d

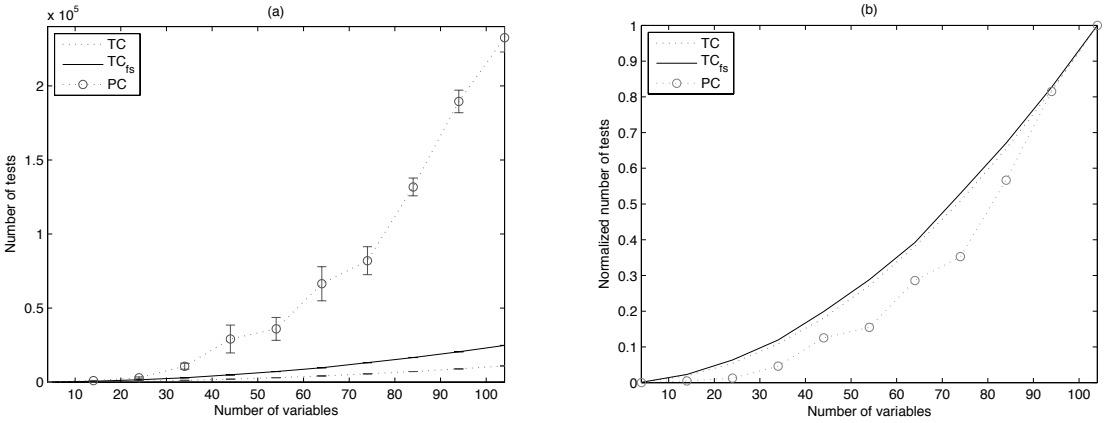

FIG. 9: Diabetes: (a) number of tests and (b) scaled number of tests as a function of d

without the exponential time complexity growth if they are not part of triangles, as in Hailfinder. In general, TC and TC_{bw} resolve all CI relations (up to married parents) in $\mathcal{O}(n^3)$ in Step 1, whereas all PC can do in $\mathcal{O}(n^3)$ is resolve CI relations with conditioning sets of cardinality 1. It is then reasonable to expect TC and TC_{bw} to scale better than PC on sparse networks where nodes have a small number of parents

PC could not be beaten on small sample sizes. It is yet an unsolved challenge for TC and TC_{bw} to handle problems where the number of variables exceeds the number of samples, as in gene expression networks, thus leading to an attempt at inverting a matrix that does not have full rank. Regularizing the covariance matrix might help make TC_{bw} more robust in this case. PC and TC are complementary in the sense that PC is preferably used with smaller sample sizes, and TC can take over more accurately and more rapidly with larger datasets.

TC_{bw} helps solving problems with TC and small datasets, but still cannot operate below the $n = d$ threshold. The exact sample size where TC_{bw} stops performing better than TC does not appear to be a simple function of the n or d but depends on the structure of the network. It would be useful to investigate when the feature selection addition of TC_{bw} becomes irrelevant. This could

lead to a criterion allowing us to merge TC and TC_{bw} into a single algorithm that knows whether or not to perform this step in order to achieve better results.

Computationally, TC_{bw} does add a degree of complexity with respect to TC, although the run times tend to be only marginally higher when compared with PC. The addition TC_{bw} does to TC is clearly to be related to feature selection. The Recursive Feature Elimination algorithm (Guyon *et al.*, 2002), for instance, works similarly for support vector machines, eliminating low-weight support vectors and reevaluating the remaining ones. Tibshirani (1994) criticizes backwards stepwise regression, which is part of TC_{bw}, arguing that the repeated tests on nonchanging data are biased, and proposes other subset selection methods, which we have not yet tested in our structure learning context.

Other algorithms retrieving the Markov blanket of single variables, and thus analogous to Step 1 of TC and TC_{bw}, include MMMB (Tsamardinos *et al.*, 2003) and HITON_MB (Aliferis *et al.*, 2003). These papers also discuss the link to feature selection. But to the best of our knowledge, none of them has been extended and applied to fully continuous datasets, and whereas they argue more about the usefulness of causal discovery techniques for feature selection, we use feature selection techniques to improve the robustness of causal discovery.

6 Conclusion

Causal discovery with continuous variables is tractable with the multivariate Gaussian assumption: We presented The TC_{bw} algorithm, which recovers the first checks for each pair of variables if their association can be accounted for by the intermediate of other variables, and if not, links them, thus determining the Markov blanket of each node. In this first pass, it uses feature selection in the first step to help address the problem of power of the statistical tests in low sample size conditions. A second pass performs a local search to detect the V-structure and orient the graph correctly.

TC_{bw} outperforms or equals the reference PC algorithm in accuracy except for small sample sizes in a fraction of its run time. In the future, we intend to investigate its behavior in the case $n < d$ and improve it.

A Appendix: Correctness Proofs

For all proofs, we assume the given dataset D is DAG-isomorphic.

Lemma A.1 *In a DAG $\mathcal{G}$, any (undirected) path π of length $\ell(\pi) > 2$ can be blocked by conditioning on any two consecutive nodes in π.*

Proof. It follows from Definition 2.2 that a path π is blocked when either at least one collider (or one of its descendants) is not in $\mathbf{S}$, or when at least one non-collider is in $\mathbf{S}$. It therefore suffices to show that conditioning on two consecutive nodes always includes a non-collider. This is the case because two consecutive colliders would require bidirected arrows, which is a structural impossibility with simple DAGs. □

Lemma A.2 *In a DAG $\mathcal{G}$, two nodes X, Y are* d*-connected given all other nodes $\mathbf{S} = \mathbf{V} \setminus \{X, Y\}$ if and only if any of the following conditions holds:*

(i) There is an arc from X to Y or from Y to X (i.e., $X \to Y$ or $X \leftarrow Y$);
(ii) X and Y have a common child Z (i.e., $X \to Z \leftarrow Y$).

Proof. We prove this by first proving an implication and then its converse.

($\Longleftarrow$) If (i) holds, then X and Y cannot be d-separated by any set. If (ii) holds, then Z is included in the conditioning set and d-connects X and Y by Definition 2.2.

($\Longrightarrow$) X and Y are d-connected given a certain conditioning set when at least one path remains open. Using the conditioning set $\mathbf{S}$, paths of length > 2 are blocked by Lemma A.1 since $\mathbf{S}$ contains all nodes on those paths. Paths of length 2 contain a mediating variable Z between X and Z; by Definition 2.2, $\mathbf{S}$ blocks them unless Z is a common child of X and Y. Paths of length 1 cannot be blocked by any conditioning set. So the two possible cases where X and Y will be d-connected are (i) or (ii). □

Corollary A.3 *Two variables X, Y are dependent given all other variables $\mathbf{S} = \mathbf{V} \setminus \{X, Y\}$ if and only if any of the following conditions holds:*

(i) X causes Y or Y causes X;
(ii) X and Y have a common effect Z.

Proof. It follows directly from Lemma A.2 due to the DAG-isomorphic structure, which ensures that there exists a DAG where CI and d-separation map one-to-one. Lemma A.2 can then be reread in terms of CI and causation instead of d-separation and arcs. □

Lemma A.4 *The subset $\mathbf{Z}$ that has the Collider Set property for the pair (X, Y) is the set of all direct common effects of X and Y and exists if and only if X is neither a direct cause nor a direct effect of Y.*

Proof. The fact that $\mathbf{Z}$ exists if and only if X is neither a direct cause nor a direct effect of Y is a direct consequence of (12), which states that X and Y can be made conditionally independent. This is in contradiction with direct causation. We now assuming that some $\mathbf{S}_{XY}$ and $\mathbf{Z}$ have been found.

($\Longrightarrow$) ($Z_i \in \mathbf{Z} \implies X \twoheadrightarrow Z_i \twoheadleftarrow Y$.) By (12) and (13), we know that each Z_i opens a dependence path between X and Y (which are independent given $\mathbf{S}_{XY}$) by conditioning on $\mathbf{S}_{XY} \cup Z_i$. By Definition 2.2, conditioning on Z_i opens a path if Z_i is either a colliding node or one of its

descendants. As, by definition, $\mathbf{Z} \subset \mathbf{Tri}(X - Y)$, we are in the first case. We conclude that Z_i is a direct effect of both X and Y.

($\Longleftarrow$) ($X \twoheadrightarrow Z_i \twoheadleftarrow Y \implies Z_i \in \mathbf{Z}$.) Note that (12) and (13) together are implied in presence of a V-structure $X \twoheadrightarrow Z_i \twoheadleftarrow Y$. Thus, a direct effect is compatible with the conditions. The fact that $\mathbf{Z}$ captures all direct effects follows from the maximization of its cardinality. □

Theorem A.5 *If the variables are jointly distributed according to a multivariate Gaussian, TC returns the CPDAG of the Markov equivalence class of the DAG representing the causal structure of the data-generating process.*

Proof. We first show that Step 1 identifies a correct skeleton $\mathcal{G}_S$ of $\mathcal{G}_0$, where each edge in $\mathcal{G}_S$ is either an arc in $\mathcal{G}_0$, indicating causation, or links two parents of a common child in $\mathcal{G}_0$, indicating a spurious link.

An edge is added in Step 1 between X and Y if we find that $\rho_{XY \cdot \mathbf{V} \setminus \{X,Y\}} \neq 0$. Owing to the multivariate Gaussian distribution, we conclude $(X \not\perp\!\!\!\perp Y \mid \mathbf{V} \setminus \{X, Y\})$. Corollary A.3 says that this implies that X causes Y or Y causes X, or that they share a common child. Therefore, each V-structure is turned into a triangle by Step 1. Step 2 then examines each link $X - Y$ part of a triangle, and by Lemma A.4, we know that if the search for a set $\mathbf{Z}$ that has the Collider Set property succeeds, there must be no link between X and Y. We know by the same lemma that this set includes all colliders for the pair (X, Y), so that all V-structures are correctly identified. Step 3 is the same as in the IC or PC algorithms; see Pearl and Verma (1991); Spirtes *et al.* (1993). □

Bibliography

B. Abramson, J. Brown, A. Murphy and R. L. Winkler, 1996. *Hailfinder: A Bayesian system for forecasting severe weather*. International Journal of Forecasting, 12:57–71. Available at: www.cs.dartmouth.edu/~donaldclass/GradAI/Readings/BayesNet_Charniak_Presentation.pdf

C. F. Aliferis, I. Tsamardinos and A. Statnikov, 2003. *HITON, A Novel Markov Blanket Algorithm for Optimal Variable Selection*. In *Proceedings of the 2003 American Medical Informatics Association (AMIA) Annual Symposium*, pp. 21–25. Available at: citeseer.ist.psu.edu/aliferis03hiton.html

S. Andreassen, R. Hovorka, J. Benn, Kristian G. Olesen and E. R. Carson, 1991. *A Model-based Approach to Insulin Adjustment*. In *Proc. of the Third Conf. on AI in Medicine*, pp. 239–248. Springer-Verlag

K. Baba, R. Shibata and M. Sibuya, 2004. *Partial correlation and conditional correlation as measures of conditional independence*. Australian & New Zealand Journal of Statistics, 46(4)

I. Beinlich, H. J. Suermondt, R. M. Chavez and G. F. Cooper, 1989. *The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks*. In *Proc. of the Second European Conf. on AI in Medicine*, pp. 247–256

N. Friedman, M. Linial, I. Nachman and D. Pe'er, 2000. *Using Bayesian networks to analyze expression data*. In *RECOMB*, pp. 127–135. Available at: citeseer.ist.psu.edu/article/friedman00using.html

I. Guyon, J. Weston, S. Barnhill and V. Vapnik, 2002. *Gene Selection for Cancer Classification using Support Vector Machines*. Machine Learning, 46(1-3):389–422. Available at: citeseer.ist.psu.edu/guyon02gene.html

G. G. Judge, R. Carter Hill, W. E. Griffiths, H. Lütkepohl and T.-C. Lee, 1988. *Introduction to the Theory and Practice of Econometrics, 2nd Edition*. Wiley

P. Leray and O. François, 2004. *BNT Structure Learning Package*. Available at: banquiseasi.insa-rouen.fr/projects/bnt-slp/

D. Margaritis, 2005. *Distribution-free learning of Bayesian network structure in continuous domains*. In *Proc. of the 20th National Conf. on AI*. Available at: www.cs.iastate.edu/~dmarg/Papers/Margaritis-AAAI05.pdf

C. Meek, 1995. *Causal inference and causal explanation with background knowledge*. In *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence*

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

J. Pearl, 2000. *Causality: Models, Reasoning, and Inference*. Cambridge University Press. Available at: singapore.cs.ucla.edu/BOOK-2K/

J. Pearl and T. Verma, 1991. *A theory of inferred causation*. In *Proc. of the Second Int. Conf. on Principles of Knowledge Representation and Reasoning*. Morgan Kaufmann. Available at: citeseer.ist.psu.edu/pearl91theory.html

A. Raveh, 1985. *On the Use of the Inverse of the Correlation Matrix in Multivariate Data Analysis*. The American Statistician, 39:39–42

R. Scheines, P. Spirtes, C. Glymour, C. Meek and T. Richardson, 1995. *The TETRAD Project: Constraint Based Aids to Causal Model Specification*. Technical report, Carnegie Mellon University, Dpt. of Philosophy. Available at: citeseer.ist.psu.edu/288318.html

P. Spirtes, C. Glymour and R. Scheines, 1993. *Causation, Prediction, and Search*, volume 81. Springer Verlag, Berlin

M. Talih, 2003. *Markov Random Fields on Time-Varying Graphs, with an Application to Portfolio Selection*. Ph.D. thesis, Hunter College

R. Tibshirani, 1994. *Regression shrinkage and selection via the lasso*. Technical report, University of Toronto. Available at: citeseer.ist.psu.edu/tibshirani94regression.html

I. Tsamardinos, C.F. Aliferis and A. Statnikov, 2003. *Time and Sample Efficient Discovery of Markov Blankets and Direct Causal Relations*. In *Proc. of the 9th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*

I. Tsamardinos, L. E. Brown and C. F. Aliferis, 2006. *The Max-Min Hill-Climbing Bayesian Network Structure Learning Algorithm*. Machine Learning. Available at: discover1.mc.vanderbilt.edu/discover/public/Publications/DSL-05-01_MergedUpdate2.pdf

S. K. M. Wong, D. Wu and T. Lin, 2002. *A Structural Characterization of DAG-Isomorphic Dependency Models*. In *Proc. of the 15th Conf. of the Canadian Society for Computational Studies of Intelligence*, pp. 195–209. Morgan Kaufmann

*In definitions, "*if*" is to be read as equivalence by definition.*