Robust Independence-Based Causal Structure Learning in Absence of Adjacency Faithfulness

Jan Lemeire   Stijn Meganck   Francesco Cartella
ETRO Department, Vrije Universiteit Brussel, Belgium
Interdisciplinary Institute for Broadband Technology (IBBT), Belgium
{*jan.lemeire, stijn.meganck, francesco.cartella*}*@vub.ac.be*

Abstract

This paper presents an extension to the Conservative PC algorithm which is able to detect violations of adjacency faithfulness under causal sufficiency and triangle faithfulness. Violations can be characterized by pseudo-independent relations and equivalent edges, both generating a pattern of conditional independencies that cannot be modeled faithfully. Both cases lead to uncertainty about specific parts of the skeleton of the causal graph. This is modeled by an f-pattern. We proved that our Very Conservative PC algorithm is able to correctly learn the f-pattern. We argue that the solution also applies for the finite sample case if we accept that only strong edges can be identified. Experiments based on simulations show that the rate of false edge removals is significantly reduced, at the expense of uncertainty on the skeleton and a higher sensitivity for accidental correlations.

1 Introduction

Independence-based algorithms for learning the causal structure from data rely on the Conditional Independencies (CIs) entailed by the system's causal structure. The *causal Markov condition* gives the CIs that follow from a causal structure that is represented by a Directed Acyclic Graph (DAG): every variable is independent of its non-effects conditional on its direct causes. All algorithms rely on a form of faithfulness. *Causal faithfulness* says that no other CIs appear in the system's probability distribution than those entailed by the causal Markov condition. Faithfulness is therefore a very convenient property: all CIs tell us something about the causal structure. Violation of faithfulness means that there are non-Markovian CIs.

The validity of causal faithfulness is supported by the 'Lebesgue measure zero argument' (Meek, 1995b), which says that the chance of randomly picking a parameterization of a Bayesian network resulting in non-Markovian CIs has measure zero. But in near-to-unfaithful situations probability distributions come infinitely close to unfaithful distributions, such that a test for independence which has to rely on a finite sample will not be able to identify the dependencies correctly. The Lebesgue measure zero argument does not hold here, since the ϵ-regions around unfaithful situations do not have Lebesgue measure zero.

(Zhang and Spirtes, 2007) showed that only in cases of *triangle unfaithfulness* violations of faithfulness are *undetectable*. This happens when the true probability distribution is not faithful to the true causal DAG, but is nonetheless faithful to some other DAG. In those cases, the CIs do not give enough evidence to learn the correct DAG. This will be discussed in more detail in the next section. We will therefore have to assume triangle faithfulness. Then, violations of faithfulness are detectable in the sense the true probability distribution is not faithful to any DAG. It means that there exist several DAGs that each explain a subset of the CIs.

(Ramsey et al., 2006) showed that we only need adjacency faithfulness and orientation faithfulness to learn the correct equivalence class. *Adjacency Faithfulness* states that any two adjacent variables do not become independent when conditioned on some other (possible empty) set of variables. It is necessary to recover the correct skeleton of the true DAG. *Orientation faithfulness* (check reference for definition) is necessary for finding the correct orientations. (Ramsey et al., 2006) extended the well-known PC algorithm to detect violations of orientation faithfulness. Violations lead to specific ambiguous parts of the DAG, in which no decision on the orientation can be taken. The Conservative PC algorithm is given in the next section. In this paper we apply the same idea for handling violations of adjacency faithfulness. They can be identified, under triangle faithfulness, by two patterns: *pseudo-independent relations* and *equivalent edges*. These patterns will lead to parts of the model in which no decision can be taken on the correct skeleton.

The following section recalls the important aspects of independence-based causal structure learning. In section 3 we analyze violations of adjacency faithfulness. Based on the identified CI patterns, the VCPC algorithm is presented and proven to be correct in section 4. Section 5 analyzes the finite sample case. Finally, the experimental results are presented in section 6.

2 Independence-Based Causal Inference

We recall the Conservative PC algorithm (CPC), see Alg. 1. $Adj(G, X)$ denotes the set of nodes adjacent to X in graph G. Single stochastic variables are denoted by capital letters, sets of variables by boldface capital letters. Step 3 consists of extensions to the original PC algorithm (Spirtes et al., 1993) in which Orientation-Faithfulness is tested (Ramsey et al., 2006). Edges of an unshielded triple, i.e. a triple $< X, Y, Z >$ for which X and Z are both adjacent to Y, but X and Z are not adjacent, are not oriented if a failure is detected, but are indicated as unfaithful, as shown in Fig. 1(a). An e-pattern is a partially-oriented DAG in which some triples are denoted as unfaithful.

Undetectable violations of faithfulness only happen by violations of the triangle faithfulness (Zhang and Spirtes, 2007) condition. It states that given a set of variables V whose true causal DAG is G, let X, Y, Z be any three variables that form a triangle in G

1. If Y is a non-collider on the path $< X, Y, Z >$, then X, Z are dependent conditional on any subset of $V \backslash \{X, Z\}$ that does not include Y.
2. If Y is a collider on the path $< X, Y, Z >$, then X, Z are dependent conditional on any subset of $V \backslash \{X, Z\}$ that includes Y

To illustrate triangle unfaithfulness, consider the DAG shown in Figure 2(b). There are 3 ways to violate triangle faithfulness for this DAG:

(TRUFF1) $X \perp\!\!\!\perp Y$ gives faithful model $X \rightarrow Z \leftarrow Y$

(TRUFF2) $Y \perp\!\!\!\perp Z$ gives faithful model $Y \rightarrow X \leftarrow Z$

(TRUFF3) $X \perp\!\!\!\perp Z | Y$ gives faithful model $X \rightarrow Y \rightarrow Z$

Besides faithfulness, *minimality* (MIN) is also a basic condition: elimination of an edge leads to a Bayesian network which violates the Markov condition. Formally:

$$\forall X, Y \in \boldsymbol{V} \text{which are adjacent in Bayesian network :}$$
$$X \not\perp\!\!\!\perp Y \mid OthPa(X–Y) \tag{1}$$

where $OthPa(X - Y)$ of edge $X - Y$ is defined as $Parents(Y) \setminus X$ if X is parent of Y, otherwise it is $Parents(X) \setminus Y$. $OthPa$ is short for 'other parents'.

3 Violation of Adjacency Faithfulness

Here we analyze unfaithfulness in the case of a perfect test for (conditional) independence. Later we will consider imperfect tests due to finite sample sizes. In all other cases than triangle unfaithfulness, there are Conditional Independencies (CIs) that make violation of faithfulness detectable. Violations of adjacency faithfulness can be identified by two patterns: pseudo-independent relations and information equivalences. Consider $X \rightarrow Y$. There are 2 kinds of violations: one in which X and Y are marginally independent and one in which they become independent when conditioned on some **Z**.

Algorithm 1 The CPC algorithm

S1 Start with the complete undirected graph U on the set of variables $\boldsymbol{V}$.

Part I ***Adjacency search***.

S2 $n = 0$;

repeat

For each pair of variables A and B that are adjacent in (the current) U, check through the subsets of $Adj(U, A) \setminus \{B\}$ and the subsets of $Adj(U, B) \setminus \{A\}$ that have exactly n variables. For all such subsets $\boldsymbol{S}$ check independency $A \perp\!\!\!\perp B \mid \boldsymbol{S}$. If independent, remove the edge between A and B in U, and record $\boldsymbol{S}$ as $Sepset(A, B)$;

$n = n + 1$;

until for each ordered pair of adjacent variables A and B, $ADJ(U, A) \setminus \{B\}$ has less than n elements.

Part II ***Orientation.***

S3 Let G be the undirected graph resulting from step S2. For each unshielded triple $\langle A, B, C \rangle$ in G, check all subsets of A's potential parents (nodes that are adjacent to A but are not A's children) and of C's potential partners:

(a) If B is NOT in any such set conditional on which A and C are independent, orient the triple as a collider: $A \rightarrow B \leftarrow C$;

(b) If B is in all such sets conditional on which A and C are independent, leave $A - B - C$ as it is , i.e., a non-collider;

(c) Otherwise, mark the triple as "unfaithful" by underlining the triple, $A \underline{-B-} C$.

S4 Execute the orientation rules given in (Meek, 1995a), but not on unfaithful triples.

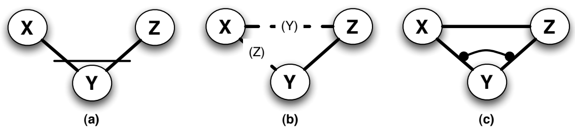

Figure 1: The three cases of uncertainty: (a) an unfaithful triple by violation of orientation faithfulness for unshielded triple $\langle A, B, C \rangle$, (b) PPIRs when $X \perp\!\!\!\perp Y$ in model $X \rightarrow Y \leftarrow Z$ and (c) equivalent edges when $X \perp\!\!\!\perp Y | Z$ in model $X \rightarrow Y \rightarrow Z$.

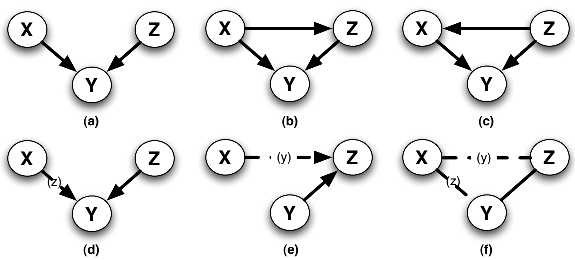

Figure 2: Marginal independency $X \perp\!\!\!\perp Y$ leads to violation of triangle faithfulness for (b) and (c). For (a) this gives a PIR, denoted in (d). This model is equivalent to (e). Both equivalent structures are denoted by PPIRs in (f).

3.1 Violation by Marginal Independence

Whenever $X \perp\!\!\!\perp Y$ for some adjacent variables X and Y, we call $X - Y$ a *Pseudo-Independent Relation* (PIR). Then, by Eq. 1, there exists at least one subset of $\boldsymbol{V}$, namely $OthPa(X - Y)$, which turns the independency into a dependency after conditioning. We call any such subset a dependency set, or *depset* for short, of X and Y, written as $depset_{XY}$. A special case in which PIRs occur is identified as *pseudo-independent models* (Xiang et al., 1996), in which three variables are pairwise independent but become dependent when conditioned on the third variable.

For not overloading the rest of the discussion we assume that for each PIR there exists a depset with one element. PIRs with larger depsets can be identified similarly, but such cases are very rare.

Assumption 1 *If X and Y are adjacent, and $X \perp\!\!\!\perp Y$, there exists a $Z \in OthPa(X–Y)$ such that $X \not\perp\!\!\!\perp Y | Z$.*

Take $Z \in OthPa(X - Y)$ which forms a depset of $X - Y$. Z is adjacent to Y. If Z would also be adjacent to X, $X \perp\!\!\!\perp Y$ is a result of triangle unfaithfulness (TRUFF1 or TRUFF2), as shown in Fig. 2 by (b) and (c). By excluding the triangle case, $X \rightarrow Y \leftarrow Z$ is an unshielded collider for which there is a $\boldsymbol{U}$ such that $X \perp\!\!\!\perp Z | \boldsymbol{U}$ and in general $X \not\perp\!\!\!\perp Z | Y, \boldsymbol{U}$. Fig. 2(a) shows such a model for which $\boldsymbol{U}$ is empty. To denote a PIR, we annotate the edge with the depset, as shown in Fig. 2(d).

A PIR implies a marginal independence and a conditional dependence. This pattern is the same as that of a v-structure. Hence, a PIR leads to two equivalent structures that can explain all CIs. Fig. 2(e) gives the same CIs as (d). We describe this pattern by connecting variables which are marginally independent but have a depset by a special edge: a Potential PIR (PPIR). A PPIR is written as $X–(Z)–Y$ and graphically denoted by a dashed edge annotated with the depset, as shown in Fig. 2(f). A PPIR can thus be a PIR or be part of a v-structure. In Sec. 4 we will see that in certain cases, a PIR can be identified from a structure with 2 PPIRs.

3.2 Violation by a non-Markovian Conditional Independence

A second violation of adjacency faithfulness happens when for adjacent variables X and Y: $X \not\perp\!\!\!\perp Y | depset_{XY}$ and there is a set $\boldsymbol{Z}$ for which $[X \perp\!\!\!\perp Y \mid \boldsymbol{Z} \cup depset_{XY}]$. The latter denotes a *strict CI*: a CI that turns into a conditional dependency for each proper subset of the conditioning set. Based on this independence, the PC algorithm would wrongly remove the edge between X and Y. We will prove that under triangle faithfulness (1) there are CIs that let us detect such false separations, and (2) the ambiguities can be represented by *equivalent edges*. We first define equivalent edges and present an example. For clarity, we omit the depset $depset_{XY}$ in the discussion.

3.2.1 Equivalent edges

The result of a false separation given by the above strict CI are 2 or more equivalent structures in which one edge can be replaced by another. We call them *equivalent edges*. Equivalent edges are linked with an arc with a bullet at each end, as shown in Fig. 1(c).

Definition 2 *Take distribution P and G a DAG not containing directed edges $X–Y$ and $Z–Y$. Two edges $X–Y$ and $Z–Y$ are called equivalent edges if and only if G is not Markovian for P and*

$$\begin{aligned} & G \cup X–Y \;\; \textit{is Markovian for } P \\ \Leftrightarrow \;\; & G \cup Z–Y \;\; \textit{is Markovian for } P \end{aligned} \tag{2}$$

Note that with $X–Y$ we denote that the edge can have both orientations. A DAG is called Markovian for a distribution if all CIs of the DAG given by the Markov condition are present in the distribution.

3.2.2 Example of Information Equivalence.

Consider the structure $Z \rightarrow X \rightarrow Y$ and the deterministic relation $X = f(Z)$. Two conditional independencies follow:

$$Z \perp\!\!\!\perp Y \mid X \;\;\&\;\; X \perp\!\!\!\perp Y \mid Z. \tag{3}$$

We call Y and Z *information equivalent* with respect to X (Lemeire, 2007). Since $X \not\perp\!\!\!\perp Y$, this a violation of the *intersection condition* (Pearl, 1988). The first equation comes from the Markov condition, the second is implied by the functional relation. X is completely determined by Z, so Z has all information about X. Knowing Z therefore renders X irrelevant for Y. Information equivalences happen when there are deterministic relations, but also under weaker conditions (Lemeire, 2007).

In the example, we have $[X \perp\!\!\!\perp Y \mid Z]$ which falsely suggests that Z separates X from Y and edge $X \rightarrow Y$ can be removed. But $Z \perp\!\!\!\perp Y \mid X$ suggests that $Y - Z$ can be removed. Removal of both edges results in a non-Markovian DAG. An ambiguity on the correct structure is a result. X or Z should be connected to Y to explain the dependencies. Structures $Z \rightarrow X \rightarrow Y$ and $X \leftarrow Z \rightarrow Y$ are equivalent given the CIs. $X - Y$ and $Z - Y$ are equivalent edges.

Concluding, CI $Z \perp\!\!\!\perp Y \mid X$ made it possible to identify a strict CI that would lead to a false separation. In the following section we present the general conditions and prove that they lead to equivalent edges.

3.2.3 Conditions for equivalent edges

When strict CI $[X \perp\!\!\!\perp Y | \boldsymbol{Z}]$ is observed, removal of $X - Y$ is only valid when **Z** is a *minimal cut set*[1] in the true graph. The following theorem gives the conditions to recognize a 'false minimal cut set'. A strict d-separation, denoted as $[X \perp Y | \boldsymbol{Z}]$, is a d-separation which gives a d-connection for any proper subset of **Z**.

Theorem 3 ***Z*** *is not a minimal cut set for X and Y in G if for one of the elements U of **Z** one of the following d-separations hold in G: ($\boldsymbol{Z'} = \boldsymbol{Z} \setminus U$ and $\boldsymbol{T} \subset V \setminus \boldsymbol{Z} \setminus \{X, Y\}$)*

1. $U \perp Y | \boldsymbol{Z'}$ *or* $U \perp X | \boldsymbol{Z'}$*;*
2. $U \perp Y | \boldsymbol{Z'}, \boldsymbol{T}$ *and* $U \perp X | \boldsymbol{Z'}, \boldsymbol{T}$*;*
3. $[U \perp Y | X, \boldsymbol{Z''}]$ *or* $[U \perp X | Y, \boldsymbol{Z''}]$ *for some* $\boldsymbol{Z''} \subset \boldsymbol{Z'}$*;*
4. $[U \perp Y | X, \boldsymbol{Z'}, \boldsymbol{T}]$ *or* $[U \perp X | Y, \boldsymbol{Z'}, \boldsymbol{T}]$.

*If none of the d-separations hold, either **Z** is a subset of a minimal cut set, or there is a $U \in \boldsymbol{Z}$ that forms a triangle or a v-structure with X and Y.*

Proof:
To be a minimal cut set, all elements of **Z** must lie on a separate path between X and Y, and all paths between X and Y must be blocked by **Z**. With path we mean an active path in terms of a d-connection. The conditions happen when **Z** is not a minimal cut set. Condition (1) or (2) hold when U is not connected to X or Y with a separate path. Condition (3) or (4) happen when U is d-connected to Y via X (by the strictness).

The last part is about what happens when none of the conditions are met. Conditions (1) or (2) guarantee that all elements of **Z** are connected with X and Y via separate paths. Next for a cut set, all paths between X and Y must be cut. If there would be an uncut path via another node, this node should be added to **Z** to form a cut set. The remaining case is when X and Y are adjacent. Then, unless U forms a triangle with X and Y, there exists a subset ***T*** which separates U from Y (or X). U can then be d-separated from Y given X, ***T*** and ***Z'*** (condition (4)) unless they form a v-structure ($U \rightarrow X \leftarrow Y$). ∎

The CIs corresponding to the d-separations of the theorem can be used to detect false separations. Condition (1) will be used in the finite sample case discussed in Section 5. The CIs corresponding to conditions (3) and (4) result in the presence of equivalent edges, as shown by the following theorem.

Theorem 4 *If G containing edges $X–Y$ and $U–Y$ is a Markovian DAG for P, $[X \perp\!\!\!\perp Y | \boldsymbol{Z'} \cup U]$ and one of the CIs corresponding to conditions (3) and (4) holds for U, then $X–Y$ and $U–Y$ are equivalent edges.*

Proof:
First we prove that $G \setminus X–Y$ is a Markovian DAG. To prove this, assume $A \not\perp\!\!\!\perp B | \boldsymbol{S}$ which holds in G but would not be represented in $G \setminus X–Y$. For this, the only active path from A to B must go via $X–Y$ and no path may exist via $U - Y$. It follows that $A \perp B | \boldsymbol{S} \cup X$ (a). A is related to X but cannot be related to U since otherwise there would be a path to Y via $U–Y$. This could only happen if U is a collider on the path between A and Y. We prove that this results in a contradiction. From $Y \rightarrow U$ follows that the path from U goes towards X to have an active path from Y to X through U (to represent the dependencies given by the strict CI). Acyclicity gives then $Y \rightarrow X$. The active path from X to A must then be pointing towards A for having an active path between A and B. This, however, creates a path from Y to A via U and X, which was excluded. Hence $A \perp U | \boldsymbol{S}$ (b). From the given CI and the CIs following from (a) and (b) follows that $A \perp\!\!\!\perp B | \boldsymbol{S}$ which results in a contradiction.

Next, by conditions (3) or (4), U could be d-separated from Y by X. But if we would also remove $U–Y$, the dependency $X \not\perp\!\!\!\perp Y | \boldsymbol{Z'}$ is not present anymore, since the path from X to Y via U is removed. The graph without both edges is thus not Markovian. The graph $G \setminus U–Y$ is also Markovian by swapping X and U in the proof. ∎

The simplest case of Condition (3), with an empty ***Z''***, was discussed in Section 3.2.2. Fig. 3 gives an exam-

[1] A cutset is a set of variables which blocks all active paths between X and Y. A cutset is minimal if no proper subset is a cutset.

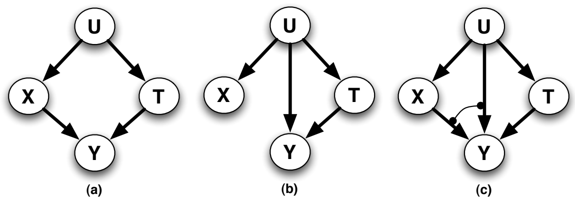

Figure 3: Example of two equivalent structures. If $X \perp\!\!\!\perp Y|U$ holds for (a) and $U \perp\!\!\!\perp Y|X, T$ for (b), than both represent the same CIs. $X - Y$ and $U - Y$ are equivalent edges, which is denoted in (c).

ple of Condition (4) with a non-empty $\boldsymbol{T}$. Assume that the non-Markovian CI $[X \perp\!\!\!\perp Y|U]$ holds. In that case, the CPC algorithm will delete edge $X - Y$. Since at that point of the algorithm Y is still connected to U, all observed dependencies are explained. But with Markovian independency $U \perp\!\!\!\perp Y|X \cup T$, deletion of edge $Y - U$ results in a model which cannot explain the dependencies. We end up with two equivalent structures: one with edge $X - Y$, the other with edge $Y - U$. Both models explain all dependencies. But the first cannot explain $X \perp\!\!\!\perp Y \mid Z$, the second cannot explain $Z \perp\!\!\!\perp Y|X \cup U$.

To simplify the rest of the discussion we will exclude the equivalences following from a condition with a non-empty **Z"** in condition (3) of the theorem. They can be treated in a similar way, but are much rarer.

Assumption 5 *For all strict independencies of the form* $[X \perp\!\!\!\perp Y|U \cup \mathbf{Z} \cup depset_{XY}]$ *(with* **Z'** $\subset$ **Z***):*

$$Y \perp\!\!\!\perp U|X \cup \mathbf{Z'} \cup depset_{XY} \Rightarrow Y \perp\!\!\!\perp U|X \cup depset_{XY}.$$

3.2.4 Relation to NPC

The necessary path condition (NPC) algorithm (Steck and Tresp, 1999) was introduced as a robust extension for the PC algorithm. It states that for each strict conditional independence $[X \perp\!\!\!\perp Y|\mathbf{Z}]$ there must exist a path between X (Y) and each $U \in \mathbf{Z}$ not crossing Y (X). This is similar to the notion described above that the minimal cutset of two variables needs to be connected to both variables. The NPC introduces the concept of ambiguous edges, which is defined as an edge whose presence depends on the absence of another. In NPC, these ambiguous regions are resolved by including a minimal number of ambiguous edges in order to satisfy a maximal number of independence relations. In our case, ambiguous regions correspond to the equivalences we find between edges. Instead of forcing them into a DAG structure, we model the ambiguity explicitly by an f-pattern. An *f-pattern* is an e-pattern augmented by edges that are denoted as PPIRs and subsets of edges denoted as equivalent. An oriented PPIR in the pattern is identified as a PIR.

3.3 Augmented Knowledge Graph

We will use an augmented knowledge graph to model the causal information. We define an *augmented* knowledge graph (AKG) (Eberhardt, 2008) as a graph containing the following relations between any two variables X and Y: $X \quad Y$, $X \rightarrow Y$, $X - Y$, $X - (\boldsymbol{S}) - Y$ and $X - (\boldsymbol{S}) \rightarrow Y$, with $\boldsymbol{S}$ a set of sets of variables. Furthermore, edges can be also related with one another either by a straight line —— or a curved line with round endpoints ●—●.

An f-pattern can be represented using an augmented knowledge graph by using the following interpretations for the different relations between variables

$X \quad Y$ Neither X nor Y are direct causes of one another.

$X \rightarrow Y$ X is a direct cause of Y.

$X - Y$ Either X is a direct cause of Y or reverse.

$X - (\boldsymbol{S}) \rightarrow Y$ There is a pseudo-independent causal relationship between X and Y with $\forall D \in \boldsymbol{S}$, D is a depset for the PIR.

$X - (\boldsymbol{S}) - Y$ There is either a pseudo-independent, direct causal or no relation between X and Y

and the following interpretations for the relation between edges:

$X - Y - Z$ The triple $< X, Y, Z >$ is unfaithful.

$X - Y - Z$ The edges $X - Y$ and $Z - Y$ are equivalent.

4 The Very Conservative PC Algorithm

The Very Conservative PC algorithm (VCPC) adds to CPC the rules of S2' to S2 (Alg. 2) and replaces Part II with Part II' (Alg. 3). Besides recording the sepsets for pairs of variables, it will also record depsets. The algorithm returns an f-pattern. When we speak about adjacencies, these special edges are also considered. Non-equivalent and non-PPIR edges are called normal edges.

Theorem 6 *(Correctness of VCPC) Consider a graph G, a JPD P generated by G and* $I(P)$*, the set of CIs of P: if minimality, triangle-faithfulness and assumptions 1 and 5 hold for P, the algorithm will, based on* $I(P)$*, return an f-pattern describing a set of DAGs that includes G. The algorithm is not trivial; it does not always return the set of all DAGs.*

Proof:
1) Assume adjacency faithfulness:
Given adjacency faithfulness, the only difference with CPC during the adjacency search is that in step S2'[II]a for each true v-structure $X \rightarrow Z \leftarrow Y$, such that $X \perp\!\!\!\perp Y$, a PPIR $X - (Z) - Y$ is added. During the first part of the orientation phase these PPIRS are temporarily removed

Algorithm 2 VCPC algorithm S2'

[I] Before testing whether $X \perp\!\!\!\perp Y | \boldsymbol{S}$ holds, check the following:

- **a** When $X - Y$ is a PPIR, add $depset_{XY}$ to ***S***.
- **b** If $X - Y$ has an equivalent edge $X - Z$ or $Y - Z$ and Z is a member of ***S***, skip the test.

[II] If the independence test returns $X \perp\!\!\!\perp Y | \boldsymbol{S}$, do the following before removing the edge:

- **a** If ***S*** is empty, look for a T in $Adj(X) \cup Adj(Y)$ for which $X \not\perp\!\!\!\perp Y \mid T$. If such a T exists, do not remove the edge, denote it as a PPIR with depset T.
- **b** If ***S*** is not empty, test for all $Z \in \boldsymbol{S}$ whether $X \perp\!\!\!\perp Z | Y \cup depset_{XY}$ and $Z \perp\!\!\!\perp Y | X \cup depset_{XY}$. If for a Z, one of both independencies hold, do not remove edge $X - Y$ and do the following. Assume the first independency is found (if the second independency holds, just swaps X and Y in the following.). (1) If $X - Z$ has been removed due to d-separation by respectively Y or X, use this edge for constructing all sets ***S*** in S2 and add the edge back to the graph and go to (3). (2) If $X - Z$ has been removed due to some other d-separation, leave edge $X - Y$ in the graph, but do not qualify it as equivalent. (3) If $Y \not\perp\!\!\!\perp Z | depset_{XY}$, denote $X - Y$ as equivalent to $X - Z$ in the graph.

to discover the v-structures. As a result in step S5a, the PPIR is removed and the correct structure is found. The correctness as well as the non-triviality follows then from the correctness and non-triviality of CPC, proven by (Ramsey et al., 2006).
2) No adjacency faithfulness:
We have to prove that no edge is deleted based on non-Markovian CIs, and that no mistakes are made during orientation.
2.1) No missing edges
A) Assume $X - Y$ in correct graph and $X \perp\!\!\!\perp Y$:
In step S2'[II]a, the algorithm looks for a variable T such that $X \not\perp\!\!\!\perp Y | T$. The existence of T follows from Minimality. Therefore the edge $X - Y$ will be replaced by a PPIR. Now, we show that this PPIR is not removed from the graph, which can only happen when there is a v-structure. If a PPIR would be removed based on the existence of a v-structure $X \rightarrow Z \leftarrow Y$ for some Z, then this indicates that the triangle faithfulness assumption is not satisfied. The removal of a PPIR in this case is

Algorithm 3 VCPC algorithm Part II'

Part II' ***Orientation.***

- Perform all of the following steps until no more edges can be oriented:

  Remove the PPIRs from G;

  Perform S3' as explained in (Ramsey et al., 2006), except that unshielded triples containing an equivalent edge are not considered;

  Perform S4 from the original algorithm on non-equivalent edges;

  Add the PPIR edges back G;

S5 Go through all PPIRs. Look for triangles consisting of normal edges and PPIRs in which for each PPIR the opposite variable in the triangle is a depset.

- **a** If the triangle contains two normal edges which form a v-structure, remove the PPIR.
- **b** If the triangle only contains one normal edge which is directed, direct the PPIR that contains the node to which the arrow of the normal edge is pointing, label the PPIR as a PIR and remove the other PPIR from the graph.
- **c** For all oriented edges $D \rightarrow A$ in G for which only A belongs to the triangle, check whether A and D form a faithful triple and a v-structure with one of the two other nodes of the triangle (as in S3 of CPC). When testing the triple A, B and D, add $depset_{AB}$ to the conditioning set of the independence tests. If a v-structure is found, orient the two triangle edges containing A towards A and delete the third triangle edge if it is a PPIR.

an immediate consequence of the triangle faithfulness assumption which dictates that the direction of one of the arcs in the triangle imposes a v-structure. We do not orient v-structures containing equivalent edges, since a v-structure based on an equivalent edge which is not in the true graph could lead to erroneous deletion of a PIR when the equivalent edge appears in a triangle with the PIR.
B) Assume $X - Y$ in correct graph and $X \perp\!\!\!\perp Y | \boldsymbol{S}, \boldsymbol{S} \neq \emptyset$:
Take $Z \in \boldsymbol{S}$. Because of triangle faithfulness, Z cannot be adjacent to both X and Y, say it is not adjacent to X. Z can then be d-separated from X which gives

$X \perp\!\!\!\perp Z|Y \cup \boldsymbol{U}$. If $\boldsymbol{U}$ is not empty, from Assumption 5 it follows that $X \perp\!\!\!\perp Z|Y$. Edge $X - Y$ is not removed. If $\boldsymbol{U}$ is empty and $X \not\perp\!\!\!\perp Z|Y$, edge $X - Y$ will be removed temporarily. It will be added back when $X \perp\!\!\!\perp Z|Y \cup \boldsymbol{U}$ is discovered at a later stage.
C) Non-triviality is a direct consequence of the deletion of an edge $X - Y$ if for $\forall S \subseteq \{X, Y\}$ the independency $X \perp\!\!\!\perp Y|S$ holds.
2.2) Correct conservative orientation:
a) Orientation in the first step of the VCPC orientation phase (II') is only based on non-equivalent edges and non-PPIRs. So the correctness of CPC proves the correctness of these orientation steps in our algorithm.
b) We do not orient any edges based on equivalent edges.
c) Both S5b and S5c trigger when there is a known orientation of a normal edge inside a triangle with (a) PPIR(s) (S5b) or when the orientation of an edge in such a triangle can be inferred (S5c). A direct consequence of triangle faithfulness is that there is a v-structure at the node of the triple which has an incoming arrow. So the correctness of orientation follows from triangle faithfulness. ∎

5 Finite Sample Case

In this section we consider the finite sample case in which the independence oracle can make errors. Let's assume that the oracle for measuring CI is based on estimating the Dependency Strength (DS) and using a threshold for deciding independency. The smaller the sample, the more the estimated DS can deviate from the true value. A higher threshold is used for smaller sample sizes so that true independencies are not misclassified as dependencies. But this implies that the weaker a (conditional) dependency is, the more likely it gets misclassified as an independency. This is especially true as the DS becomes lower than the threshold. The oracle will only detect dependencies that are sufficiently strong. The following three cases should be considered.

5.1 Weak edges.

An edge $X{-}Y$ with a small $DS(X;Y)$ can still have a high $DS(X;Y|Z)$ when conditioned on one of the other parents, as is shown by the PIRs. A PIR still contains a lot of information, despite the marginal independence. Our extensions overcome missing PIRs or quasi-PIRs (edges that look like PIRs due to the finite sample size). On the other hand, if both $DS(X;Y)$ and $DS(X;Y|OthPa(X{-}Y))$ are small, we cannot overcome overlooking such edges, which we call *weak edges*. Limited data gives limited precision.

5.2 Near-to-unfaithfulness.

In general, dependencies with a low DS lead to near-to-unfaithful situations. Faithful distributions can come infinitely close to the unfaithful cases. This leads to the same CI patterns as in the unfaithful cases.

5.3 Weakening by conditioning.

A third way in which limited samples disrupt the learning is that an increased cardinality of the conditioning set reduces the robustness of most independence tests (Spirtes et al., 1993, p.116). We call this effect 'weakening by conditioning', which results in strict CIs not corresponding to minimal cut sets. They can be detected by the CIs corresponding to the conditions of Theorem 3.

6 Experimental results

To illustrate the adequacy of our extensions, simulations were performed on linear Gaussian and binary models. Experiments were performed on 100 randomly selected DAGs with d nodes and d edges, where d is randomly chosen between 5 and 25. For each such graph, a random structural equation model was constructed by selecting edge coefficients randomly uniformly from $[0.1, 1] \cup [-1, -0.1]$ and the variance of the disturbance terms was chosen randomly from $[0.01, 1]$. A random data set of 1000 cases was simulated for each of the models, to which the PC, CPC and VCPC algorithms were applied with depth 2 and significance level $\alpha = 0.05$ for each independence test based on Fisher's Z transformation of partial correlation. The output graph was compared to the Markov equivalence class (MEC) of the true DAG. Similar experiments were performed with Bayesian networks defined over a set of binary variables and randomly chosen conditional probabilities. The Chi-Square test was used as independence test.

The table on the next page shows the outcomes averaged over all experiments and relative to the number of nodes (percentages). Correct edges are the edges of the MEC of the true graph that appear as normal edges in the f-pattern. PPIRs and equivalent edges in the f-pattern are counted as ambiguous edges. False negative edges are edges in the MEC that do not appear in the f-pattern, not as a normal edge and not as an ambiguous edge. Weak edges are false negatives whose nodes are marginally independent and independent conditional on the other parents. False positive edges appear as normal edges in the f-pattern, but not in the MEC. If the nodes of a false positive are not d-connected in the MEC, they are classified as 'not connected'.

The learning performance of the orientation is evaluated by looking at edges appearing in both the MEC and the f-pattern. Edges having the same orientations in both are counted as correct orientations, when not oriented in

	PC	CPC	VCPC
Edges			
Correct	76.7	76.2	77.9
Ambiguous	0.0	0.0	74.4
False negatives	**23.2**	**23.8**	**8.1**
Weak	3.7	4.5	3.6
False positives	**4.1**	**4.3**	**11.8**
Not connected	2.9	3.1	9.3
Orientations			
Correct	25.3	29.8	36.6
Ambiguous	3.6	16.5	47.6
Wrong	**19.9**	**2.1**	**3.7**
False positives	**15.1**	**2.9**	**3.9**

both or only in f-pattern as ambiguous. Wrong orientations appear as oriented in both, but in the opposite direction. False positives are arrowheads appearing in the f-pattern but not in the MEC.

The results show that the difference between the PC and CPC lies clearly in the reduction of the false positive arrowheads, although we notice an increase in false negatives. The VCPC algorithm clearly reduces the number of false negative edges. If we consider that weak edges cannot be identified, the performance gain is even more drastic. By subtracting the number weak edges from the false negatives, the number of false negatives drops from 19.5%/19.3% for PC/CPC to 4.5% for VCPC. This drop is at the expense of ambiguous edges and more false positives. The latter can be explained by *accidental correlations*.

Accidental correlations lead to false negative independence tests - the oracle qualifies a Markovian CI as dependent due to accidentally-correlated data. VCPC is conservative about dependencies, it will not remove edges if there is no alternative path to explain a dependency. Take nodes that are not d-connected in the true graph ('not connected' in the table), but are accidently correlated. If this accidental correlation is above the threshold, the oracle will qualify it as a dependency. In the following steps, when conditioning happens on other variables, the weakening-by-conditioning effect will bring the measured dependency strength below the threshold and remove the 'accidental' edge. This happens with PC and CPC.

Finally, the experiments showed that the standard deviation for the false positive and negative edges is almost as high as the average, which points to a high performance fluctuation from one experiment to another.

7 Conclusions

We cannot rely on adjacency faithfulness when constructing robust learning algorithms. We showed that under triangle faithfulness, violations can be detected by two patterns: potential pseudo-independent relations (PPIRs) and equivalent edges. Based on both patterns, a set of DAGs can be identified that are indistinguishable from the perspective of the CIs. Just like the Conservative PC algorithm detects and treats failures of orientation-faithfulness, our Very Conservative PC algorithm detects violations of adjacency-faithfulness.

In the finite sample case, weak conditional dependencies can be wrongly classified as CIs by the oracle. This leads to near-to-unfaithful cases, weakening by conditioning and weak edges. The two first are treated, missing weak edges should be accepted. Since weak edges and triangle unfaithfulness cannot be detected, we believe that this analysis shows the natural bounds of what can reliably be learned under causal sufficiency.

References

Frederick Eberhardt. 2008. Almost optimal intervention sets for causal discovery. In *UAI*, pages 161–168.

Jan Lemeire. 2007. *Learning Causal Models of Multivariate Systems and the Value of it for the Performance Modeling of Computer Programs*. Ph.D. thesis, Vrije Universiteit Brussel.

Christopher Meek. 1995a. Causal inference and causal explanation with background knowledge. In *Procs of UAI-1995*, pages 403–41.

Christopher Meek. 1995b. Strong completeness and faithfulness in Bayesian networks. In *Procs of UAI-1995*, pages 411–418.

Judea Pearl. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA, Morgan Kaufman Publishers.

Joseph Ramsey, Jiji Zhang, and Peter Spirtes. 2006. Adjacency-faithfulness and conservative causal inference. In *Procs of UAI-2006*, pages 401–408.

Peter Spirtes, Clark Glymour, and Richard Scheines. 1993. *Causation, Prediction, and Search*. Springer Verlag, 2nd edition.

Harald Steck and Volker Tresp. 1999. Bayesian belief networks for data mining. In *Procs of the 2nd Workshop on Data Mining und Data Warehousing*.

Yang Xiang, S. K. Wong, and N. Cercone. 1996. Critical remarks on single link search in learning belief networks. In *Procs of UAI-1996*, pages 564–571.

Jiji Zhang and Peter Spirtes. 2007. Detection of unfaithfulness and robust causal inference. In *Procs of the LSE-Pitt Conference: Confirmation, Induction and Science, London*.