Learning Causal Structure from Overlapping Variable Sets

Sofia Triantafillou **Ioannis Tsamardinos** **Ioannis G. Tollis**
Computer Science Department, University of Crete and
Institute of Computer Science, FORTH, Hellas

Abstract

We present an algorithm name cSAT+ for learning the causal structure in a domain from datasets measuring different variable sets. The algorithm outputs a graph with edges corresponding to all possible pairwise causal relations between two variables, named Pairwise Causal Graph (PCG). Examples of interesting inferences include the induction of the absence or presence of some causal relation between two variables never measured together. cSAT+ converts the problem to a series of SAT problems, obtaining leverage from the efficiency of state-of-the-art solvers. In our empirical evaluation, it is shown to outperform ION, the first algorithm solving a similar but more general problem, by two orders of magnitude.

1 Introduction

Modern data-analysis fields, such as machine learning and statistics, for the most part study the isolated analysis of a single dataset. Several data analysis subfields have developed methods to integratively analyze heterogeneous datasets such a Multi-Task Learning, Transfer Learning, and Meta-Analysis to name a few, arguably with limited success. In recent work (Tsamardinos and Triantafillou, 2010), we argue that the reason for the inability of the methods to soundly co-analyze a larger set of datasets is due to the prevalence of association (correlation) as the conceptual cornerstone of data analysis. Instead, co-analyzing heterogeneous datasets is feasible if the analysis is based on causal models. By making additional assumptions about the connection of causality and estimable quantities such as probability distributions, the observed associations (dependencies and independencies) in one dataset, constrain the causal mechanism that fits other datasets. This allows the integrative causal analysis of datasets obtained under different experimental conditions, different sampling designs, and datasets measuring different, possibly intersecting variable sets. In this paper, we focus on this latter problem.

We assumed a single (unknown) causal mechanism over the observed variables $\mathbf{O} = \cup_i \mathbf{O}_i$ and latent variables $\mathbf{L}$ (observed in no available dataset) generates the data. It is also assumed that this mechanism can be represented by a faithful Causal Bayesian Network, and so the dependencies and independencies of the marginal over $\mathbf{O}$ are captured by a Maximal Ancestral Graph (MAG) and the m-separation criterion (Richardson and Spirtes, 2002).

The proposed algorithm accepts the set of datasets as well as a causal query. The query regards whether a particular causal hypothesis could be true in the unknown generating MAG, e.g, whether A directly causes B. The algorithm then tries to identify a MAG whose marginal over each $\mathbf{O}_i$ fits the corresponding dataset and simultaneously satisfies the query. If this is deemed impossible, then the hypothesis that edge $A \rightarrow B$ is in the generating MAG is rejected, otherwise a satisfying MAG model is returned. The key idea of the algorithm is to convert the problem of identifying a model with the desired properties to a SAT problem and gain leverage from the research and technology of SAT solvers. By repeatedly invoking the algorithm for each possible type of causal relation between two variables in $\mathbf{O}$ one can construct a new type of causal graph, that we name Pairwise Causal Graph (PCG). PCGs summarize the pairwise structural uncertainty among variables. Other types of queries to the algorithm are possible, e.g, testing whether A is indirectly causing B.

In our empirical evaluation, we show that the algorithm performs a significant number of non-trivial inferences, compared to the inductions made by analyzing each dataset in isolation. Interesting cases include the inference that two variables are not associated

Appearing in Proceedings of the 13^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2010, Chia Laguna Resort, Sardinia, Italy. Volume 9 of JMLR: W&CP 9.

(have no inducing path or latent confounder between them) even though they are never measured together in the data. Similarly, that two variables never measured together are directly causally related. Given the wealth of different datasets in certain domains, such as medicine and biology, such inferences have important ramifications for the future of data analysis and causal discovery.

We compare the proposed algorithm against ION (Tillman et al., 2008), the first algorithm that can make causal inferences from datasets defined over different variable sets. ION however, produces all data-generating PAGs (equivalent classes of MAGs) and so it is a more general algorithm: for any given query, one can go through the list of PAGs and check whether the query holds for any models. If one is interested in testing the causal possibilities of a single edge, the proposed algorithms obtains a speed-up of about two orders of magnitude over ION and scales to larger problems. Finally, we also present preprocessing steps that lead to significant efficiency gains and point to future directions for gaining efficiency. The code is downloadable from http://www.mensxmachina.org/ .

2 Preliminaries

Maximal Ancestral Graphs is a type of model that represents both causal relations among a set of variables $\mathbf{O}$ as well as probabilistic properties, such as conditional independencies (independence model). The causal semantics of an edge $A \rightarrow B$ imply that A is probabilistically causing B, i.e., a manipulation of A results in a change of the distribution of B. Edges $A \leftrightarrow B$ imply that there is a latent (not in $\mathbf{O}$) common cause of A and B. Under certain conditions, the independencies implied by the model are given by a graphical criterion called m-separation, defined below. Thus, MAGs are useful in probabilistic reasoning both when observing a system and under manipulation (control) of the system. A desired property of MAGs is that they are closed under marginalization: any marginal distribution can be represented by a MAG. MAGs can also represent the presence of selection bias, but this is out of the scope of the present paper. We present the key theory of MAGs.

A path in $\mathcal{G}$ is a sequence of distinct vertices $\langle V_0, V_1, \ldots, V_n \rangle$ s.t for $0 \leq i < n$, V_i and V_{i+1} are adjacent in G. A path from V_0 to V_n is *directed* if for $0 \leq i < n$, V_i is a parent V_{i+1}. X is called an *ancestor* of Y and Y a descendant of X if $X = Y$ or there is a directed path from X to Y in $\mathcal{G}$. $\mathbf{An}_{\mathcal{G}}(X)$ is used to denote the set of ancestors of node X in $\mathcal{G}$. A *directed cycle* in $\mathcal{G}$ occurs when $X \rightarrow Y \in E$ and $Y \in \mathbf{An}_{\mathcal{G}}(X)$. An almost directed cycle in $\mathcal{G}$ occurs when $X \leftrightarrow Y \in E$ and $Y \in \mathbf{An}_{\mathcal{G}}(X)$.

Definition 2.1 *A mixed graph is* **ancestral** *if the graph does not contain any directed or almost directed cycles.*

Given a path $p = \langle V_0, V_1, \ldots, V_n \rangle$, node V_i, $i \in 1, 2, \ldots, n$ is a *collider* on p if both edges incident to V_i have an arrowhead towards V_i. We also say that triple (V_{i-1}, V_i, V_{i+1}) forms a collider. Otherwise V_i is called a *non-collider* on p. A triple of nodes (X, Y, W) is called *unshielded* if X is adjacent to Y, Y is adjacent to W, and X is not adjacent to W. A path $p = \langle X, \ldots, W, V, Y \rangle$ is called a *dicriminating* path for V if X is not adjacent to Y, and every vertex between X and Y is a collider on p and a parent of Y.

The criterion of m-separation will lead to a graphical way of determining the probabilistic properties stemming from the causal semantics of the graph:

Definition 2.2 *In a mixed graph $\mathcal{G} = (V, E)$, a path p between A and B is* ***m-connecting*** *relative to (condition to) a set of vertices $\mathbf{Z}$, $\mathbf{Z} \subseteq \mathbf{V} \setminus \{A, B\}$ if*

1. *Every non-collider on p is not a member of $\mathbf{Z}$.*
2. *Every collider on the path is an ancestor of some member of $\mathbf{Z}$.*

A and B are said to be ***m-separated*** *by $\mathbf{Z}$ if there is no m-connecting path between A and B relative to $\mathbf{Z}$. Otherwise, we say they are* ***m-connected*** *given $\mathbf{Z}$. We denote the m-separation of A and B given $\mathbf{Z}$ as $MSep(A; B|\mathbf{Z})$.*

For the remainder of the paper, we make the assumption that a single (unknown) causal mechanism over the observed variables $\mathbf{O} = \cup_i \mathbf{O}_i$ and possibly other latent variables $\mathbf{L}$ (observed in no available dataset) generates the data. It is also assumed that this mechanism can be represented by a faithful Causal Bayesian Network. Causal Bayesian Networks assume Causal Markov Condition holds: every variable is independent of its non-effects given its direct causes in $\mathbf{O} \cup \mathbf{L}$. The condition directly implies certain independencies hold and entails other independencies. If all and only the independencies among $\mathbf{O} \cup \mathbf{L}$ are entailed by the Markov Condition, the network follows the Faithfulness Condition. In the remainder of the paper we assume that both conditions hold. Notice that, the two conditions connect causality with estimable properties such as independencies. We define the set $\mathcal{J}$ of all pairwise conditional independencies $X \perp Y|\mathbf{Z}$:

$$\mathcal{J} \equiv \{\langle X, Y|\mathbf{Z}\rangle, \text{s.t.}, X \perp Y|\mathbf{Z} \text{ and } \{X\}, \{Y\}, \mathbf{Z} \subseteq \mathbf{O}\}$$

Similarly, we define the set $\mathcal{J}(\mathcal{G})$ of all m-separations $MSep(X; Y|\mathbf{Z})$ in an ancestral graph $\mathcal{G}$. Under the

state assumptions, the independencies correspond to m-separations:

$$\mathcal{J} = \mathcal{J}(\mathcal{G})$$

This result connects the probability distribution with the corresponding graph.

Definition 2.3 *An ancestral graph $\mathcal{G}$ is called* maximal *if for every pair of non-adjacent vertices (X, Y), there is a (possibly empty) set $\mathbf{Z}$, $X, Y \notin \mathbf{Z}$ such that $\langle X, Y, \mathbf{Z}\rangle \in \mathcal{J}(\mathcal{G})$.*

Hence, every non-adjacency in a Maximal Ancestral Graph (MAG) implies at least one corresponding independence between the non adjacent variables in the independence model. Maximality can also be expressed in term of a special kind of paths, called inducing paths, as described in (Richardson and Spirtes, 2002).

Definition 2.4 *In an ancestral graph $\mathcal{G} = (V, E)$, a path π between X and Y is **inducing** relative to (with respect to) a set of vertices $\mathbf{L}$, $\mathbf{L} \subseteq \mathbf{V} \setminus \{X, Y\}$ if every collider on π is an ancestor of X or Y, and every non-collider is in $\mathbf{L}$. If $\mathbf{L} = \emptyset$, π is called a primitive inducing path.*

It is entailed by the definition of m-separation that if p is an inducing path between X and Y relative to $\mathbf{L}$, then there is no subset $\mathbf{Z}$ of $\mathbf{L}$ that renders them conditionally independent: $\forall \mathbf{Z} \subseteq \mathbf{L} \setminus \{X, Y\}, \langle X, Y|\mathbf{Z}\rangle \notin \mathcal{J}$. Thus, in a maximal ancestral graph there is no primitive inducing path between two variables if and only if the variables are non-adjacent. We denote the independence model stemming from marginalizing over variables $\mathbf{L}$ as $\mathcal{J}[_{\mathbf{L}}$, i.e.

$$\mathcal{J}[_{\mathbf{L}} \equiv \{\langle X, Y|\mathbf{Z}\rangle \in \mathcal{J}; (X \cup Y \cup \mathbf{Z}) \cap \mathbf{L} = \emptyset\}$$

A simple graphical transformation for a MAG G with independence model $\mathcal{J}$ exists that provides a unique MAG $G[_{\mathbf{L}}$ that represents the causal ancestral relations and the independence model $\mathcal{J}[_{\mathbf{L}}$ after marginalizing out variables in $\mathbf{L}$. Formally,

Definition 2.5 *Graph $\mathcal{G}[_L$ has vertex set $\mathbf{V} \setminus \mathbf{L}$, and edges defined as follows: If X, Y are s.t. , $\forall \mathbf{Z} \subseteq \mathbf{V} \setminus (\mathbf{L} \cup \{X, Y\})$, $\langle X, Y|\mathbf{Z}\rangle \notin \mathcal{J}(\mathcal{G})$ and*

$$\begin{array}{lll} X \notin \mathbf{An}_{\mathcal{G}}(Y); Y \notin \mathbf{An}_{\mathcal{G}}(X) & & X \leftrightarrow Y \\ X \in \mathbf{An}_{\mathcal{G}}(Y); Y \notin \mathbf{An}_{\mathcal{G}}(X) & \textit{then} & X \rightarrow Y \quad \text{in } \mathcal{G}[_L \\ X \notin \mathbf{An}_{\mathcal{G}}(Y); Y \in \mathbf{An}_{\mathcal{G}}(X) & & X \leftarrow Y \end{array}$$

The following result have been proved in (Richardson and Spirtes, 2002):

Theorem 2.1 *If $\mathcal{G}$ is a MAG over $\mathbf{V}$, and $\mathbf{L} \subset \mathbf{V}$, then $\mathcal{G})[_{\mathbf{L}}$ is also MAG and*

$$\mathcal{J}(\mathcal{G})[_{\mathbf{L}} = \mathcal{J}(\mathcal{G}[_L)$$

Different MAGs encode different causal information, but may share the same independence models. Such statistically indistinguishable MAGs define a Markov equivalence class. The following result has been proved in (Spirtes and Richardson, 1997):

Proposition 2.1 *Two MAGs over the same vertex set are Markov equivalent if and only if:*

1. *They share the same edges*
2. *They share the same unshielded colliders*
3. *if a path p is discriminating for a vertex V in both graphs, V is a collider on the path on one graph if and only if it is a collider on the path on the other.*

A *Partial Ancestral Graph* is a graph containing (up to) three kinds of endpoints: arrowhead ($>$), tail ($-$), and circle ($\circ$), and represents a MAG Markov equivalence class in the following manner: It has the same adjacencies as any member of the equivalence class, and every non-circle endpoint is invariant in any member of the equivalence class. Circle endpoints correspond to uncertainties; the definitions of paths are extended with the prefix *possible* to denote that there is a configuration of the uncertainties in the path rendering the path ancestral, inducing or m-connecting. For example if $X \circ\!\!-\!\!\circ Y \circ\!\!\rightarrow W$, $\langle X, Y, W\rangle$ is a possible ancestral path from X to W, but not a possible ancestral path from W to X. Example PAGs are shown in Figure 1(b-c). FCI (Spirtes et al., 2000; Zhang, 2008) is an asymptotically correct algorithm which outputs a PAG over a set of variables $\mathbf{V}$ when given access to an independence model over $\mathbf{V}$.

3 An Algorithm for Finding Consistent MAGs

We assume that we are given access to a set of independence models $\{\mathcal{J}_i\}_{i=1}^K$ over corresponding subsets of variables $\mathbf{O}_i$. For example, we may be given datasets from which the independencies in the models can be determined by the use of statistical tests. For the purposes of this paper, we assume an oracle of conditional independencies and do not deal with possibility of statistical errors. We define the problem of identifying a MAG consistent with all $\mathcal{J}_i$ where we use the notation $\overline{\mathbf{O}_i} \equiv \mathbf{O} \setminus \mathbf{O}_i$.

Definition 3.1 (Find Consistent MAG) *Given independence models $\{\mathcal{J}_i\}_{i=1}^K$ over subsets of variables $\mathbf{O}_i$, induce a MAG $\mathcal{M}$ s.t., for all i*

$$\mathcal{J}(M[_{\overline{\mathbf{O}_i}}) = \mathcal{J}_i$$

We now present an algorithm that converts the problem above to a satisfiability problem, s.t. a MAG is consistent iff corresponds to a truth-setting assignment to the SAT instance that does not induce directed cycles. First, we discuss how the problem can be recast in graph-theoretic terms. Let $\mathcal{P}_i$ be the PAG representing the Markov equivalence class of all MAGs consistent with the independence model $\mathcal{J}_i$. $\mathcal{P}_i$ can be constructed with a sound and complete algorithm such as FCI. We can thus recast the problem above as identifying a MAG $\mathcal{M}$ such that, $M[_{\overline{\mathbf{O}_i}} \in \mathcal{P}_i$, for all i (abusing the notation to denote with $\mathcal{P}_i$ the equivalence class).

In turn, the above observation implies that MAG $\mathcal{M}$ should be such that $\mathcal{M}[_{\overline{\mathbf{O}_i}}$ has the same edges (adjacencies), the same unshielded colliders and the same discriminating colliders(colliders discriminated by discriminating paths) as $\mathcal{P}_i$. If an edge is missing from $\mathcal{P}_i$ due to independence $\langle X, Y|\mathbf{Z}\rangle$, this implies X and Y should be m-separated given $\mathbf{Z}$ in $\mathcal{M}$. Similarly, if an edge is present in $\mathcal{P}_i$, there should be an inducing path relative to $\overline{\mathbf{O}_i}$ in $\mathcal{M}$. These constraints on the graph are converted to a SAT instance. The primitive variables in this instance correspond to edge existence and endpoint orientations.

1. $edge(X, Y) = edge(Y, X)$ is true when X is adjacent to Y.
2. $arrowhead(X, Y)$ is true when X is into Y

Based on the above we also define:

1. $collider(X, Y, Z)$ where X, Y, Z is an ordered triple, is true when X, Y, Z forms a collider. Obviously $collider(X, Y, Z) \Leftrightarrow collider(Z, Y, X)$
2. $ancestor(X, \mathbf{Y})$ is true when X is an ancestor of some $Y \in \mathbf{Y}$.
3. $ancestral(X_1, X_2, \ldots, X_n)$ is true when $\langle X_1, X_2, \ldots, X_n\rangle$ is a directed path (possibly empty) from X_1 to X_n.
4. $inducing(X_1, X_2, \ldots, X_n, \mathbf{L})$ is true when $\langle X_1, X_2, \ldots, X_n\rangle$ is an inducing path between X_1 and X_n w.r.t $\mathbf{L}$.
5. $mconnecting(X_1, X_2, \ldots, X_n, \mathbf{Z})$ is true when $\langle X_1, X_2, \ldots, X_n\rangle$ is an m-connecting path between X_1 and X_n condition to $\mathbf{Z}$

The above concepts can be expressed in terms of the primitive components of the graph, based on the definitions of m-separation, inducing path, and standard graph concepts (see (Triantafillou, 2010) for full details). Any truth-setting assignment to the primitive variables uniquely determines a mixed graph.

Algorithm 1: Find Consistent MAG (FCM)

Input: PAGs $\{\mathcal{P}_i\}_{i=1}^N$ over variables $\{\mathbf{O}_i\}_{i=1}^N$
Input: Causal Query Φ_q
Result: MAG $\mathcal{K}$

```
K ← InitializeGraph ({P_i}_{i=1}^N);
Φ_c ← Φ_q;
Φ_c ← GenerateConstraints ({P_i}_{i=1}^N, K);
repeat
    L ← solveSAT (Φ_c);
    if L = ∅ then
        return ∅
    end
    K ← makeChanges (K, L);
    for each (almost) directed cycle in K do
        add constraints to Φ_c preventing cycle
    end
until K has no (almost) directed cycles ;
return K;
```

Function `InitializeGraph`($\{\mathcal{P}_i\}_{i=1}^N$)

```
K ← complete unoriented graph over O = ⋃_i O_i ;
Transfer non-adjacencies and orientations from all P_i to K ;
Mark all edges as uncertain;
return K
```

The algorithm is shown in Algorithm 1. It accepts a set of PAGs $\{\mathcal{P}_i\}_{i=1}^N$ and a causal query Φ_q that we assume null for the moment and returns a MAG $\mathcal{K}$ consistent will all marginal PAGs. Alternatively, it can accept a set of independence models or datasets and use FCI to induce the PAGs. The algorithm begins with the complete unoriented graph over all variables $\mathbf{O} = \bigcup_i \mathbf{O}_i$. Missing edges and endpoint orientations are transferred from each $\mathcal{P}_i$. At this point, a SAT variable is implicitly introduced for every non-missing edge and every unoriented edge-endpoint. Subsequently, all the constraints induced by each PAG are added to a CNF formula Φ_c by calling `GenerateConstraints`. A SAT solution is sought by calling `solveSAT` and if no solution is found, an empty graph is returned. Otherwise, the truth-value of the primitive variables is applied on the graph by calling `makeChanges`. The resulting graph $\mathcal{K}$ is checked for directed or almost directed cycles and if none is found, K is the sought after MAG and returned. Otherwise, the cycle is detected and a SAT clause forbidding the cycle is appended in the SAT instance. This is repeated until a solution is found, or not other SAT solutions exist. We note

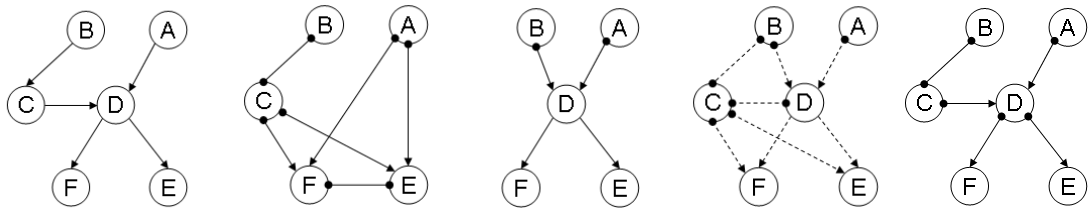


Figure 1: An example of algorithm cSAT.(a) The underlying causal network.(b)The PAG when D is not observed.(c) The PAG when C is not observed.(d) The initial graph resulting from Function `InitializeGraph`. (e) The PCG returned by cSAT, which coincides with the PAG over the union of variables. Notice that there is a solid edge between variables C and D, even though they have never been measured together; also, edge (C, F) is missing even though it is present in graph (c), the only one that includes both variables.

that all cycle-preventing constraints could be added in advance so that any truth-assignment to the SAT instance corresponds to a consistent MAG; however, in anecdotal preliminary experiments, we determined that post-adding constraints is more time efficient.

Function `GenerateConstraints`$(\{\mathcal{P}_i\}_{i=1}^N)$

1 **for** *all* $\mathcal{P}_i$ *over* $\mathbf{O}_i$ **do**
2 **if** X, Y *are adjacent in* $\mathcal{P}_i$ **then**
3 $\Phi_c \leftarrow \Phi_c \wedge$ `AdjacencyConstraints`$(X, Y, \overline{\mathbf{O}_i}, \mathcal{K})$
4 **end**
5 **else if** X, Y *are not adjacent in* $\mathcal{P}_i$ **then**
6 $\Phi_c \leftarrow \Phi_c \wedge$ `MSeparationConstraints`$(X, Y, S_{XY}, \mathcal{K})$
7 **end**
8 **end**
9 $\Phi_c \rightarrow \Phi_c \wedge$ `AdditionalConstraints`$(\mathcal{K})$;
10 **return** Φ_c ;

In function `GenerateConstraints` (lines 2-4) every edge that has been encountered in at least one $\mathcal{P}_i$ is considered. For each such edge, a set of boolean constraints are introduced to ensure that in the integrated model over $\mathbf{O}$, either the edge is present or a relative inducing path w.r.t. to $\overline{\mathbf{O}_i}$ is present. Function `adjacencyConstraints` describes the generation of these constraints. The inducing paths attempting to substitute for an edge are required to be non-primitive (imposed by $\neg inducing(path, \emptyset)$) (line 8). This way the resulting graph maintains the maximality property. Procedure `possibleInducingPaths`$(X, Y, \mathbf{L}, \mathcal{K})$ returns all paths between X and Y in $\mathcal{K}$ for which there exists an assignment to the primitive variables that makes them inducing w.r.t. $\mathbf{L}$.

In function `GenerateConstraints` (lines 5-8), edges that have been eliminated from some $\mathcal{P}_i$ are considered. A missing edge (X, Y) corresponds to at least one conditional independence $\langle X, Y | S_{X,Y} \rangle$ found by FCI when inducing $\mathcal{P}_i$. The separating sets $S_{X,Y}$ are cached during execution of the FCI algorithm so they are not rediscovered. For every missing edge (X, Y), a set of constraints is added to the formula requiring that no m-connecting path exists in $\mathcal{K}$ between (X, Y) condition on $S_{X,Y}$. Notice that for each missing edge only one m-separation is imposed on $\mathcal{K}$; however, these are all the m-separations identified by FCI when inducing $\mathcal{P}_i$. Given that the latter algorithm is sound and complete, these are enough to entail all the same independencies in $\mathcal{K}[\overline{\mathbf{O}_i}$ as in $\mathcal{K}$. Functions `possibleMConnectingPaths` and

Function `AdjacencyConstraints`$(X, Y, \mathbf{L}, \mathcal{K})$

1 $\Phi_c \leftarrow \emptyset$;
2 paths$\leftarrow$`possibleInducingPaths`$(X, Y, \mathbf{L}, \mathcal{K})$;
3 **for** *each path* $\in$ *paths* **do**
4 $\Phi_c \leftarrow \Phi_c \vee inducing(path, \mathbf{L})$;
5 **end**
6 **if** X, Y *are adjacent in* $\mathcal{K}$ **then**
7 $\Phi_c \leftarrow \Phi_c \vee edge(X, Y)$;
8 **for** *each path* $\in$ *paths* **do**
9 $\Phi_c \leftarrow \Phi_c \wedge (edge(X, Y) \vee \neg inducing(path, \emptyset))$;
10 **end**
11 **end**
12 **return** *CNF*(Φ_c)

MSeparationConstraints$(X, Y, \mathbf{Z}, \mathcal{K})$

1 $\Phi_c \leftarrow \varnothing$;
2 paths$\leftarrow$`possibleMConnectingPaths`$(X, Y, \mathbf{Z}, \mathcal{K})$;
3 **for** *each path* $\in$ *paths* **do**
4 $\Phi_c \leftarrow \Phi_c \wedge \neg mconnecting(path, \mathbf{Z})$;
5 **end**
6 **return** *CNF*(Φ_c)

`possibleInducingPaths` are implemented as an extension to the algorithm in (Neapolitan, 2003) for determining d-separation.

Finally, procedure `AdditionalConstraints` adds the constraint that all edges in $\mathcal{K}$ should have at least one arrowhead; in addition, that marked definite non-colliders in any $\mathcal{K}$ are also non-colliders in $\mathcal{K}$. Procedure CNF converts a boolean formula to an equisatisfiable formula in Conjunctive Normal Form of size linear to the size of the input; we used the algorithm in (Jackson and Sheridan, 2005) to avoid an exponential explosion of clauses during conversion. Figure 1 provides an illustrative example; any MAG output by the algorithm belongs in the equivalence class of PAG shown in Figure 1e (in general, there are typically multiple PAGs consistent with all independence models). We prove the algorithm is sound and complete in the sense that it returns a MAG if and only if it is consistent will all $\mathcal{P}_i$ and the query (the proof is in (Triantafillou, 2010)).

4 The Causal SAT Algorithm

Under the assumptions stated and when $\Phi_q = \emptyset$, FCM will always return a consistent MAG that fits the data. However, one may be interested in testing whether there exists a MAG with a specific property, e.g., the existence or absence of an edge or the presence of a path (indirect causation). Properties that can be expressed using the primitive graph terms of *edge* and *arrowhead* can be tested by augmenting the SAT formula with corresponding constraints, passed as parameter Φ_q to the algorithm. If FCM returns null, there is no fitting MAG with the given property. For example, the causal query $\Phi_q = edge(X,Y) \wedge arrowhead(X,Y) \wedge \neg arrowhead(Y,X)$ will fail if X cannot be a parent of Y. There may be many MAGs fitting the marginal distributions provided, however, they may belong to different Markov equivalence classes, i.e., are represented by different PAGs. There is currently no known compact representation of this set of solutions. One way to succinctly present causal information is to capture all possible pairwise causal relations among variables:

Definition 4.1 *Let $\{\mathcal{P}_i\}_{i=1}^N$ be a set of partial ancestral graphs over $\mathbf{O}_{i\,i=1}^N$. A Pairwise Causal Graph $\mathcal{U}$ is a partially oriented mixed graph over $\bigcup_i \mathbf{O}_i$ with two kinds of edges (--, —) and three kinds of endpoints(>, -, ∘) with the following properties:*

1. *X --Y in $\mathcal{U}$ if X is adjacent to Y in at least one $\mathcal{M}$ consistent with all $\mathcal{P}_i$.*
2. *X —Y in $\mathcal{U}$ if X is adjacent to Y in every $\mathcal{M}$ consistent with all $\mathcal{P}_i$.*
3. *X is into (out of) Y in $\mathcal{U}$ if X is into (out of) Y in every $\mathcal{M}$ consistent with all $\mathcal{P}_i$, where X and Y are adjacent.*

Algorithm 6: cSAT+

Input: $\{\mathcal{P}_i\}_{i=1}^N$
Output: $\mathcal{U}$, the Pairwise Causal Graph over $\mathbf{O} = \bigcup_{i=1}^N \mathbf{O}_i$

```
1  U ← InitializeGraph (P);
2  repeat
3  |  Apply preprocessing steps 1 and 2;
4  until no step is applicable ;
5  for every edge X,Y in U do
6  |  if FCM ({P_i}_{i=1}^N, edge(X,Y)) == ∅ then
7  |  |  Remove edge from U
8  |  end
9  |  else if FCM ({P_i}_{i=1}^N, ¬edge(X,Y) == ∅ then
10 |  |  Mark edge as solid
11 |  end
12 end
13 for every unoriented endpoint X ∗···∘ Y in U do
14 |  if FCM
   |  ({P_i}_{i=1}^N, edge(X,Y) ∧ arrowhead(X,Y)) == ∅
   |  then
15 |  |  Orient X out of Y
16 |  end
17 |  else if FCM
   |  ({P_i}_{i=1}^N, edge(X,Y) ∧ ¬arrowhead(X,Y)) == ∅
   |  then
18 |  |  Orient X into Y
19 |  end
20 end
21 return U
```

The presence of dashed edge in a PCG denotes that there exists at least one possible data-generating MAG where this edge is present, whereas solid edges represent adjacencies that are present in every possible data-generating MAG. Similarly, an oriented endpoint corresponds to an invariant orientation in every possible data-generating MAG where the respective edge exists. For non oriented endpoints (denoted by circles) there are consistent MAGs with either endpoint configuration. Pairwise Causal Graphs semantically represent the causal possibilities between two variables, and cannot be used to produce Maximal Ancestral Graphs consistent with the data without further reasoning. An example of a PCG is shown in Figure 1e.

The **Causal SAT** algorithm (cSAT) repeatedly invokes FCM with a causal query for every uncertainty present after initializing graph $\mathcal{K}$. Each rejected query is imposed on $\mathcal{K}$ which is returned as the output Pairwise Causal Graph.

4.1 Speeding up the algorithm

The size of the SAT problem depends on the number of possible inducing and m-connecting paths after initialization. Removing or orienting edges reduces the number of paths and improves efficiency. We have identified two preprocessing steps that perform limited, polynomial-time reasoning to remove and orient certain edges in advance.

Proposition 4.1 (Preprocessing Step 1) *If* $X \leftarrow Y$ *in* $\mathcal{P}_i$, *and* $MSep(X, W|\mathbf{Z})$ *in* $\mathcal{P}_j$, $Y \notin \mathbf{O}_j$ *with* $\mathbf{Z} \cap \overline{\mathbf{O}_i} = \emptyset$, *remove* $Y \circ\!\!-\!\!-\circ W$ *from* $\mathcal{U}$.

Proposition 4.2 (Preprocessing Step 2) *If* $X \rightarrow Y$ *in* $\mathcal{P}_i$, *and* $MSep(X, W|\mathbf{Z})$ *in* $\mathcal{P}_j$ *with* $\mathbf{Z} \cap \overline{\mathbf{O}_i} = \emptyset$, *orient* $Y \circ\!\!-\!\!-\circ W$ *as* $Y \leftarrow\!\!-\!\!- \circ W$ *in* $\mathcal{U}$.

The idea is that the presence of the removed edge (preprocessing step 1) or rejected orientation (preprocessing step 2) would be m-connecting the variables that have been found independent. Algorithm **cSAT+** checks every triplet (X, Y, Z) to apply the steps before generating the constraints (proofs of correctness in (Triantafillou, 2010)).

5 Results

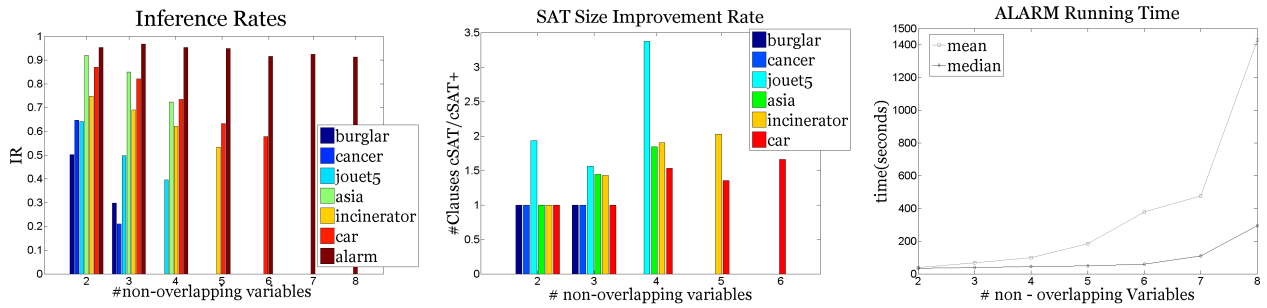
Evaluation of Inference Capabilities. We empirically quantify the inference capability of cSAT+ on 7 common networks in the literature (see (Triantafillou, 2010) for full details). The networks are named Cancer(5 variables), Burglar(5 variables), Jouet5(7 variables), Asia(8 variables), Incinerator(9 variables), Car(12 variables), and ALARM (37 variables). For each network, the variable set is partitioned in two disjoint subsets of common and non-common variable set. The non-common variable set is then randomly partitioned into two disjoint non-empty subsets. The resulting sets are joined with the common set to form two overlapping sets. FCI algorithm with an oracle of conditional independence is used to create the PAGs over the two subsets which are then fed to cSAT+. This procedure is iterated for non-common sets of size 2 to half of the variables of every network (except for ALARM), and repeated for 20 (cancer and burglar networks) or 50 (jouet5, asia, incinerator and car) random sets. MINISAT2.0 (Eén and Sörensson, 2004) is used to solve the SAT instances and the PCG corresponding to each example was constructed.

We try to quantify the number of inferences as follows. For an edge in a PCG we count the number of models it admits: from a minimum of 1 if the edge is absent or fully oriented and solid, to a maximum of 4 if the edge is fully unoriented and dashed. We quantify the total structural uncertainty conveyed by the graph $\mathcal{G}$ as the sum of this number over all edges, denoted by $SU(\mathcal{G})$. Let $\mathcal{K}_0$, $\mathcal{U}$, $\mathcal{P}$ be the graphs returned by `InitializeGraph`, cSAT+, and FCI over the complete set of variables. These correspond to the structures learnt by analyzing the datasets in isolation and trivially conjoining the results, by integrative analysis, and the optimal structure inferred when all variables are measured together. The inference rate $IR = \frac{SU(\mathcal{K}_0)-SU(\mathcal{U})}{SU(\mathcal{K}_0)-SU(\mathcal{P})}$ denotes the percentage of inferences made compared to $\mathcal{P}$ scaled to [0,1]. IR is zero when no additional inferences are made and 1 when the structure coincides with $\mathcal{P}$, the structure learnt from all variables. The figure clearly shows the inferential advantages gained by integrative analysis: most inference rates are significantly higher than zero. Somewhat surprisingly, for the larger network (ALARM) the inference rates remain above 0.9 for all sizes of set differences between the variable sets tested: the results are close to the graph learnt given all 37 variables.

Preprocessing Improves Efficiency. We have tested cSAT+ (with preprocessing) against cSAT. Without preprocessing the algorithm does not scale to the ALARM network. For smaller networks, Figure 2b presents the ratio of the median SAT clauses created by the two algorithms. The results show that the polynomial-time preprocessing step reduces the size of the SAT problem. In some cases, the number of clauses is reduced by a factor of 3 or more. The smaller SAT problems translate to overall computational efficiency; Table 1 shows the median times spent by each algorithm.

Comparison with ION. We compare the algorithm with ION, a similar but more general algorithm. Table 1 presents the timing results, where the missing values are the cases where ION runs out of memory in all iterations. ION never scales to problems where the set difference between the variable sets is more than 3 variables. ION enumerates all fitting PAGs taking up to 2 orders of magnitude more time than cSAT+.

Scaling Up. The proposed algorithm cSAT+ allows us to scale up integrative causal analysis to non-trivial problems, such as the ALARM network. Using the same design as for the other networks, we generate two random variable sets, with the size of non-overlapping variables ranging from 2 to 8. We repeat the experiment with 100 random variable splits for each parameter value and present mean and median execution time in Figure 2c. It is interesting to note that this difference increases with the number of non-overlapping variables. This implies that the execution time greatly depends on the graph structures of the marginal distributions and so certain large problems may still be solved efficiently.



(a) (b) (c)

Figure 2: Experimental Results

Table 1: Execution times in seconds for the ION, cSAT, and cSAT+.

Net	JOUET5(7v)			ASIA(8v)			INCINERATOR(9v)				CAR(12v)				
SetDiff	2	3	4	2	3	4	2	3	4	5	2	3	4	5	6
Ion	0.37	2.30	-	10.47	64.06	-	24.30	-	-	-	8.61	330.28	-	-	-
cSAT	0.37	0.62	1.57	0.45	0.88	1.81	0.71	1.35	3.38	7.97	0.40	0.67	1.15	4.14	2.69
cSAT+	0.32	0.54	0.98	0.37	0.63	1.32	0.65	1.15	2.48	4.62	0.38	0.61	0.91	2.19	2.32

6 Conclusions and Discussion

We present an algorithm for learning the causal structure in a domain from datasets measuring different variables sets, named cSAT+. The algorithm improves efficiency over ION by two orders of magnitude for the larger problems. We also introduce the Pairwise Causal Graph (PCG) to summarize the structural uncertainty of the solution set. Our results show that a large number of additional inferences is possible when datasets are integratively analyzed, compared to analysis in isolation. The existence or absence of association between variables never measured together is possibly inferred; surprisingly, the absence of edge (X, Y) may also be inferred even when (X, Y) is present in all marginal structures measuring both X and Y. These preliminary results are encouraging to further develop the methods to scale to larger and more realistic sizes, and in situations where there is no perfect knowledge of independencies (statistical errors).

Acknowledgements

We are grateful to Robert Tillman for providing us with an implementation of ION, and the VPH NoE GA no 223920 and REACTION GA 248590 EU projects, and the University of Crete for partial funding.

References

Eén, N. and Sörensson, N. (2004). An extensible SAT-solver. In *Theory and Applications of Satisfiability Testing*, pages 333–336.

Jackson, P. and Sheridan, D. (2005). Clause form conversions for boolean circuits. In *SAT 2004*.

Neapolitan, R. E. (2003). *Learning Bayesian Networks*. Prentice Hall.

Richardson, T. and Spirtes, P. (2002). Ancestral graph Markov models. *Annals of Statistics*, 30(4):962–1030.

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

Spirtes, P. and Richardson, T. (1997). A polynomial time algorithm for determining DAG equivalence in the presence of latent variables and selection bias.

Tillman, R. E., Danks, D., and Glymour, C. (2008). Integrating locally learned causal structures with overlapping variables. In *NIPS*.

Triantafillou, S. (2010). Learning causal structure from overlapping variable sets. Master's thesis, University of Crete, Heraklion.

Tsamardinos, I. and Triantafillou, S. (2010). The possibility of integrative causal analysis: Learning from different datasets and studies. *Journal of Engineering Intelligent Systems*, to appear.

Zhang, J. (2008). On the completeness of orientation rules for causal discovery in the presence of latent confounders and selection bias. *Artif. Intell.*, 172(16-17):1873–1896.