University of Crete
Computer Science Department

Learning Causal Structure from Overlapping Variable Sets

Sofia Triantafillou

Master of Science Thesis

Heraklion, February 2010.

Contents

1 Introduction **11**

1.1 Motivation . . . 11

1.2 Related Work . . . 13

2 Causal Bayesian Networks **17**

2.1 Basic Definitions . . . 17

2.2 The Causal Markov and Causal Faithfulness Conditions . . . 17

2.3 D-Separation . . . 19

2.4 Markov Equivalence . . . 20

2.5 Causal Insufficiency . . . 20

2.6 Factorization and Manipulation . . . 21

3 Maximal Ancestral Graphs **23**

3.1 Basic Definitions . . . 23

3.1.1 Independence Models . . . 23

3.1.2 Mixed Graphs . . . 24

3.1.3 Paths and Ancestors . . . 24

3.2 M-Separation . . . 24

3.2.1 Ancestral Graphs . . . 24

3.2.2 Maximality . . . 25

3.3 Marginalization . . . 26

3.3.1 Inducing Paths . . . 27

3.3.2 Marginalizing Ancestral Graphs . . . 28

3.4 Markov Equivalence . . . 28

3.4.1 Partially Oriented Ancestral Graphs . . . 31

3.5 Algorithm *Fast Causal Inference* . . . 31

4 Algorithm cSAT+ **37**

4.1 Problem Definition . . . 37

4.2 Preliminaries . . . 38

4.3 Algorithm *Find Consistent MAG* . . . 40

4.3.1 Initialization . . . 41

4.3.2 Generation of the constraints . . . 43

4.3.3 Soundness and Completeness . . . 46

4.4 The Pairwise Causal Graph 48
4.5 Algorithm cSAT+ 50

5 Results 53
5.1 Evaluation of Inference Capabilities 53
5.2 Preprocessing Improves Efficiency 54
5.3 Comparison with ION 54
5.4 Scaling Up 55
5.5 Conclusions and Future Work 55

List of Figures

1.1 Reichenbach's Common Cause Principle 12
1.2 The water sprinkler network 14

2.1 Conditioning on colliders and their descendants 19
2.2 Statistical Indistinguishability 20
2.3 A confounder reveals its presence 21

3.1 Mixed graphs that are not ancestral. 25
3.2 Maximal and non-maximal ancestral graphs 26
3.3 Markov Equivalent MAGs 29
3.4 Discriminating path 29
3.5 Markov equivalence and discriminating colliders 30
3.6 A Partially Oriented Ancestral Graph 32
3.7 FCI Orientation Rules $\mathcal{R}1 - \mathcal{R}3$ 34
3.8 FCI Orientation Rules $\mathcal{R}8 - \mathcal{R}10$ 34

4.1 Edges that appear due to partial observation 42
4.2 An example run of algorithm cSAT. 49
4.3 Preprocessing steps 1 and 2 51

5.1 Running time for cSAT+ on 7 Networks. 54
5.2 SAT size Improvement ratio 55
5.3 Running time for cSAT+ on ALARM Network. 56

Abstract

Modern data-analysis methods are typically applicable to a single dataset. In particularly, they cannot integratively analyze datasets containing different, but overlapping, sets of variables. We show that by employing *causal* models instead of models based on the concept of association alone, it is possible to make additional interesting inferences by integrative analysis than by independent analysis of each dataset. Specifically, we assume that all datasets are generated by the a single overarching causal model representable by a Maximal Ancestral Graph; Maximal Ancestral Graphs are a class of graphical independence models designed to model marginal distributions and cope with causal insufficiency (latent confounding variables). We rigorously define the problem of identifying one or all causal models that simultaneously fit the available data. We propose a novel algorithm *FCM* that converts this problem to a SAT formula whose solutions correspond to all plausible causal models. We also introduce a new kind of graphical model, the *Pairwise Causal Graph* (*PCG*), that succinctly summarizes all pairwise causal relations among the variables. Based on *FCM*, we propose *cSAT+*, an algorithm that outputs the *PCG* when given a set of datasets and prove that this algorithm is sound and complete in the absence of statistical errors. In our empirical evaluation on simulated datasets, we show that the integrative analysis using *cSAT+* makes more sound causal inferences than by analyzing the datasets in isolation. Examples of interesting inferences include the induction of the absence or the presence of some kind of causal relation between two variables never measured together. The latter observation has significant ramifications for data analysis as it implies that additional causal relations may be inferred from already available datasets, without further studies. We also show empirically that *cSAT+* outperforms ION by two orders of magnitude, the first algorithm solving a similar but more general problem, and scales to larger-sized problems than ION.

Περίληψη

Οι σύγχρονες μέθοδοι ανάλυσης δεδομένων εφαρμόζονται συνήθως σε ένα μεμονωμένο σύνολο δεδομένων. Συγκεκριμένα, αδυνατούν να ενοποιήσουν σύνολα δεδομένων που περιέχουν διαφορετικά αλλά αλληλεπικαλυπτόμενα σύνολα μεταβλητών. Υιοθετώντας αιτιακά μοντέλα αντί για τα συνήθη μοντέλα που βασίζονται αποκλειστικά σε συσχετίσεις, είναι δυνατή η εξαγωγή επιπλέον συμπερασμάτων μέσω της ενοποιημένης ανάλυσης, σε σύγκριση με την απομονωμένη ανάλυση κάθε συνόλου δεδομένων.

Υποθέτουμε ότι όλα τα σύνολα δεδομένων έχουν παραχθεί από ένα λανθάνον αιτιακό μοντέλο, που μπορεί να αναπαρασταθεί από έναν Μέγιστο Προγονικό Γράφο. Οι Μέγιστοι Προγονικοί Γράφοι είναι ένα είδος γραφικών μοντέλων ανεξαρτησίας σχεδιασμένο να μοντελοποιεί περιθώριες κατανομές και καταστάσεις αιτιακής ανεπαρκειας. (μεταβλητές που αποτελούν κρυμμένες κοινές αιτίες) .

Ορίζουμε το πρόβλημα της ταυτοποίησης ενός ή όλων των αιτιακών μοντέλων συμφωνούν με όλα τα διαθέσιμα σύνολα δεδομένων. Προτείνουμε έναν αλγόριθμο, τον FCM, που μετατρέπει το πρόβλημα σε μια λογική πρόταση SAT της οποίας οι αληθοτιμές αντιστοιχούν στα εύλογα αιτιακά μοντέλα. Ορίζουμε επίσης ένα νέο γραφικό μοντέλο, τον Διμερή Αιτιακό Γράφο, που συνοψίζει τις πιθανές διμερείς αιτιακές σχέσεις μεταξύ των μεταβλητών. Βασιζόμενοι στον FCM,προτείνουμε τον cSAT+, έναν αλγόριθμο που παράγει τον Διμερή Αιτιακό Γράφο από ένα σύνολο συνόλων δεδομένων, και αποδεικνύουμε ότι ο αλγόριθμος είναι σωστός και πλήρης όταν δεν υπάρχουν στατιστικά σφάλματα.

Στην εμπειρική ανάλυση, σε προσομοιωμένα σύνολα δεδομένων, δείχνουμε ότι η ενοποιημένη ανάλυση με τον cSAT+ επιτρέπει περισσότερα συμπεράσματα σε σχέση με την απομονωμένη ανάλυση των συνόλων δεδομένων. Παραδείγματα τέτοιων ενδιαφέροντων συμπερασμάτων είναι η επαγωγή της απουσίας ή της παρουσίας άμεσης αιτιότητας ανάμεσα σε μεταβλητές που δεν έχουν μετρηθεί μαζί. Αυτή η παρατήρηση έχει σημαντικές επιπτώσεις στην ανάλυση δεδομένων, αφού δείχνει ότι επιπλέον αιτιακές σχέσεις μπορούν να συναχθούν από δεδομένα ήδη διαθέσιμα, χωρίς τη διεξαγωγή επιπλέον πειραμάτων.

Chapter 1

Introduction

1.1 Motivation

Modern data-analysis fields, such as machine learning and statistics, for the most part study the isolated analysis of a single dataset. The results are published in the literature and researchers manually synthesize this knowledge in their heads. Obviously, this procedure blatantly underutilizes the available data and is limited by our cognitive capacities.

We argue that the reason for the inability of the methods to encompass 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.

For the most part, the only acceptable means of inducing causal relations has been by controlled experiments and specifically, by Randomized Controlled Trials [10]. However, controlled experiments are often impossible, costly, or unethical. In addition, it ignores and wastes data that have been collected without randomization (observational data). The motto "correlation does not imply causation" is imprinted in all students of statistics. But can there be correlation without causation? Set aside coincidence (or magic), the relationship between correlation and causation can be summarized in the *common cause principle*[21]: "Every enduring correlation between events is explained by a direct causal connection, an indirect causal connection or a (direct or indirect) common cause".

A simple example illustrating the aftermaths of that principle is the shown in Figure 1.1. Between any two variables, all five *causal* structures depicted in Figure 1.1 are possible. However, an observed correlation between them rules out (d) and (e), whereas the lack of such correlation rules out the first three

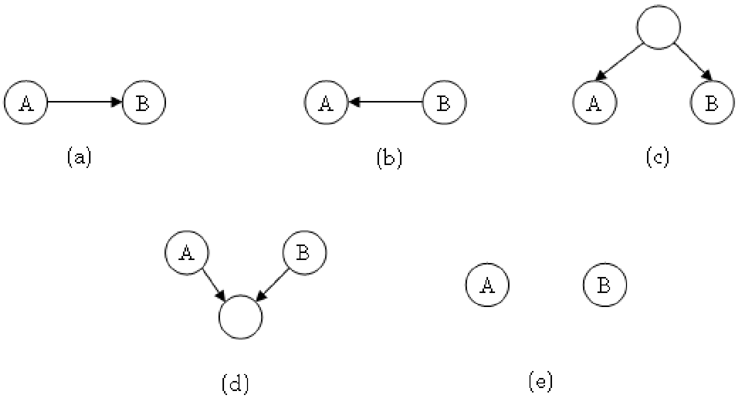


Figure 1.1: An example of Reichenbach's Common Cause Principle. An observed correlation between A and B rules out structures (d) and (e).

models.

Causal claims come with a lot of baggage; they are related to claims of effect, prediction of intervention, counterfactuals, as well as an implied temporal status. The notion of casuality alone is a matter of long philosophical debate. Causal modeling, however, has been taking place in people's minds forever, and most of our actions and decisions is based on the *effects* we believe these actions will have. The deterministic notion of causality demands that the effects invariantly follow their causes. Causal modeling, is often used in situations with uncertainty (Smoking *causes* lung cancer; however, not all smokers get lung cancer). Probability theory is used in most of the domains where causal modelling is used. Thus, antecedents are assumed to make the consequences *more likely.*

Ever since Reichenbach's common cause principle, research on causal discovery has made significant progress. Several theories have been developed attempting to *infer* causal relations from observational data. Regardless of the *type* of causal relations and *approach* considered, the process of inferring causal models from observational data aims to construct a possible model that explains the observed associations. In the following section we briefly review causal frameworks and discovery algorithms.

Since causal inference has become possible, its benefits (compared to associative analysis) can be used to utilize the numerous available datasets examining relevant issues. State-of the art methods are limited to studying a single dataset in order to test a specific hypothesis or predict a certain variable. With a growing number of such datasets becoming publicly available, the following question arises: is it possible to automatize the procedure of synthesis of available knowledge?

In this work, we focus on the problem of combining causal structures that correspond to overlapping variable sets. The algorithms we propose are motivated by the fact that in many domains, especially in biology, similar experiments testing similar hypotheses are very common. The results of these experiments, however, are only manually combined (in human minds). We argue that, due to the transitivity of causal relations, all this information can be integratively analyzed, saving a lot of time, money and effort invested in further experiments.

1.2 Related Work

For many years, the only acceptable means of causal inference was ad hoc experiments. However, recently,the advances in graphical models and computational tools has facilitated the development of a well-founded logic to describe causality, and applicable algorithms to infer causal characteristics.

Several causal frameworks have been proposed that follow different assumptions and are suitable for certain types of data. Granger causality [12] refers to causal relations among time series. A time series X is said to Granger-cause Y if it can be shown, usually through a series of F-tests on lagged values of X (and with lagged values of Y also known), that those X values provide statistically significant information about future values of Y. Dynamic causal modelling has recently been introduced to model causality in dynamical systems, where causality lies in the set of differential equations defining the system. Structural Equation Models [18] are causal models that emphasize on the influential power variables have on each other. They consist of a causal diagram, like the one depicted in Figure 1.2 and a set of independent equations that quantify the strength of the relationships. SEMs are Markovian models, i.e. every variable is considered to be influenced only by its (Markovian) parents, and an error term. Causal Bayesian networks, which we will explore in more detail in the following chapter, are probabilistic causal models. They consist of a directed causal graph that does not allow circles, and a joint probability distribution.

Learning Bayesian networks has proved to be NP-complete[4]. Nevertheless, the literature of discovery algorithms is extensive. We will only mention some of them. Algorithms SGS[24], CI[20], IC[32] and PC[24] are some constraint-based algorithms, where conditional independence tests are used to extract the DAG's skeleton and then orientation rules are applied. K2 and and GES[5] are score-based algorithm, where an initial network is scored according to a metric and changes are propagated to achieve the best score. Hybrid algorithms combining the two approaches exist, such as CB[23], K2[6] and MMHC[29].

The possibility of causal inference has opened the door to another possibility: That of integrative causal analysis. Causal relationships offer this possibility due to their transitive nature. However, it is only in the last decades that interest on causal discovery has grown, and not much work has been done on co-analyzing data.

Several data analysis subfields have developed methods to interactively analyze heterogeneous datasets such a Multi-Task Learning[3], Transfer Learning[17],

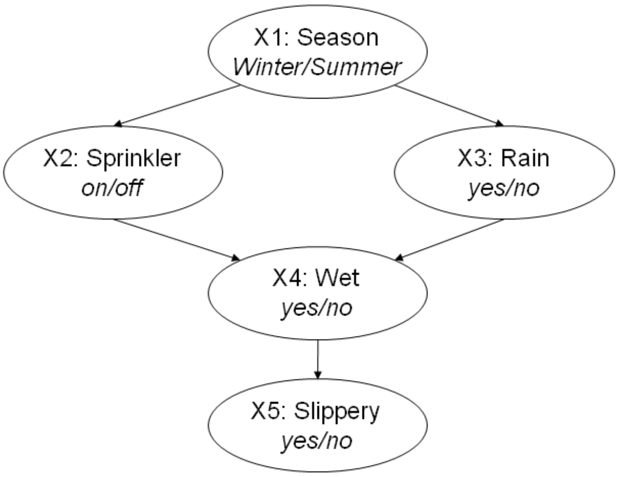


Figure 1.2: The causal diagram for a slippery pavement. Season affects the probability of rain or open water sprinklers, which both cause wet therefore slippery pavements.

and Meta-Analysis[16] to name a few. The first two address to studies under the same sampling and experimental conditions on the same sets of variables, whereas transfer learning deals with issues of transferring the results or experience of learning in one domain to a different domain. None of the aforementioned techniques involves (yet) causal modelling. Methods for combining datasets that are not identically distributed have also been proposed [28], [30].

In this work, we present a method for fitting a causal model over a set of variables $\mathbf{O}$ that have been observed in overlapping subsets $\mathbf{O}_i$. SPLR [7] is an algorithm tha utilizes locally learned structures to speed up the discovery of the joint structure. It assumes, however, that a joint dataset is also given. ION Algorithm introduced in [8] integrates locally learned structures , *Partially Oriented Inducing Path Graphs*[1] in particular, and produces a combined structure including uncertain edges. ION was then alternated in [27]. This modified version takes as input a set of distributed causal models over overlapping variable sets, and outputs the set of possible data-generating causal models. Although the objective of our algorithm is not exactly the same as that of ION, tasks performed by our algorithm could, in principle, be performed by ION, and vice versa. For example, if one should want to fit a model with specific characteristics to a series of datasets as described above, they could either use algorithm FCM (described in section 4.3), or use ION and check if a model with the specified properties is among the output models. Conversely, if one would desire to find all possible models, they could either use ION or iteratively use FCM. A com-

[1] Partially Oriented Inducing Path graphs is an alternate approach to PAGs(defined in the next chapter)

parison to ION can be found in chapter 5. Both algorithms work with *Partially Oriented Ancestral graphs*. These graphical models are somewhat a "representative" of a Markov equivalence class of Maximal Ancestral graphs, that are suitable for modeling marginal distributions. The next chapter examines the properties of these graphs.

Chapter 2

Causal Bayesian Networks

Several graphical models have been used to represent the conditional independencies that holds in a distribution. Probabilistic properties are linked to graphical properties through the Markov Condition, which can be translated for every graphical model to a series of Markov Properties. Before proceeding to define the Markov Condition, we must first revise some basic graph theoretic terms.

2.1 Basic Definitions

A *directed graph* $\mathcal{G}$ is a pair $(\mathbf{V}, \mathbf{E})$, where $\mathbf{V}$ is a finite set of elements called *nodes* and $\mathbf{E}$ is a set of ordered pairs of distinct elements of $\mathbf{V}$. Elements in $\mathbf{E}$ are called *edges.* If there is an edge from X to Y, where $X, Y \in \mathbf{V}$ then X and Y are *adjacent.* A path in G is a sequence of distinct vertices $\langle V_0, V_1, \dots, V_n \rangle$ s.t $\forall i, \quad 0 \leq i < n$, V_i and V_{i+1}, are adjacent in G, and no vertex appears more than once in the sequence.. A path from V_0 to V_n is *directed* if $\forall 0 \leq i < n$, 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 G. $\mathbf{Pa}_{\mathcal{G}}(X), \mathbf{Ch}_{\mathcal{G}}(X), \mathbf{An}_{\mathcal{G}}(X)$ and $\mathbf{De}_{\mathcal{G}}(X)$ are used to denote the set of parents, children, ancestors and descendants of node X in G, respectively. A *directed cycle* in G occurs when $X \rightarrow Y \in E$ and $Y \in \mathbf{An}_{\mathcal{G}}(X)$. A directed graph is called *acyclic* (DAG) if it contains no directed cycles.

2.2 The Causal Markov and Causal Faithfulness Conditions

Definition 2.2.1 (Markov condition) *A joint probability distribution P over a set of random variables $\mathbf{V}$ and a DAG $D = (\mathbf{V}, \mathbf{E})$ are said to satisfy (with each other) the **Markov Condition** if every variable in $\mathbf{V}$ is independent of its non-descendants condition on (all) its parents.*

A *Bayesian Network* is a a probability distribution P with a DAG D, where (P, D) satisfy the Markov Condition. A causal DAG is a DAG where every edge denoted a *direct cause*[1].

Definition 2.2.2 (Causal Markov Condition) *A joint probability distribution P over a set of random variables $\mathbf{V}$ and a causal DAG $D = (\mathbf{V}, \mathbf{E})$ are said to satisfy (with each other) the **Causal Markov Condition** if every variable in $\mathbf{V}$ is independent of its non-effects condition on its direct causes.*

A *causal Bayesian Network* is a a probability distribution P with a DAG D, where (P, D) satisfy the Causal Markov Condition. Even though the Causal Markov Condition is a subject of long philosophical debate, inference algorithms inferring causal structures from data usually assume it holds. We refer to this assumption as the *Causal Markov Assumption.* However, the Causal Markov Assumption is meaningful only for *causally sufficient systems.*

Definition 2.2.3 (Causal Sufficiency) *Given a set of variables $\mathbf{V}$, and two variables $X, Y \in \mathbf{V}$ a variable W is called a common direct cause of X and Y relative to V if W is a direct cause of X and also a direct cause of Y relative to $\mathbf{V} \cup \{W\}$. V is said to be causally sufficient if for every pair of variables $X, Y \in \mathbf{V}$, every common direct cause of X and Y relative to $\mathbf{V}$ is also a member of $\mathbf{V}$.*

In a causal sufficient system, represented by a DAG, the Markov Condition specifies a set of independence relations. These relations reflect the *local Markov property*, which coincides with $\mathcal{CMC}$ for DAGs. Faithfulness Condition ensures that the independencies entailed by the $\mathcal{CMC}$ are the only independence relationships among variables in $\mathbf{V}$.

Definition 2.2.4 (Causal Faithfulness Condition) *A joint probability distribution P over a set of random variables $\mathbf{V}$ and a causal DAG $D = (\mathbf{V}, \mathbf{E})$ satisfy the **Causal Faithfulness Condition** if every and only if every conditional independence relation true in P is entailed by the Causal Markov Condition applied to D. D and P are called faithful.*

Given a causal graph, with the aforementioned prerequisites holding, the causal Markov Condition defines a set of independence relations, stemming from the application of the $\mathcal{CMC}$. However, not all entailed independencies are obvious from this criterion, A more general criterion allows us to read all holding independencies directly from the DAG [19]. It is based on the notion of collider. A collider is a vertex with two incoming edges. The triple of vertices (the middle node with its two neighbors) is also referred to as a collider, and we also say that a vertex X *forms* a collider on any path where its preceding and following vertex are into X. For Bayesian networks, colliders have a special

[1] Direct in the sense that there is no mediating cause with respect to the variables participating in the DAG

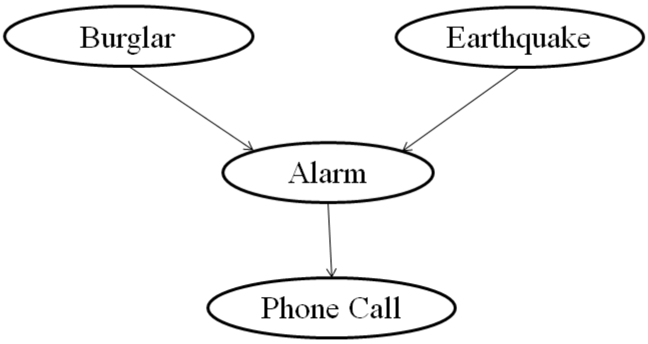


Figure 2.1: A small Bayesian network representing a home alarm. When the alarm is set off, information on *Earthquake* affects the probability of *Burglar*.

meaning. A collider on the path is considered to ”block” the flow of information, whereas instantiation of the collider allows information to flow. Assume that the network shown in Figure 2.2, represents a house alarm that goes off when a burglar enters the house, but is also set off by earthquakes sometimes . Earthquakes and burglars are unconditionally independent. Assume now, that the alarm is set off, causing your neighbor to call you in order to inform you. At the same time, you learn that an earthquake takes place. This information on the earthquake lowers the probability that a burglar has broken into your house. Thus, *Burglar* and *Earthquake* are dependent condition to *Alarm*, and *Phone call.* Thus, conditioning on a collider or a descendant of a collider, ”opens” a path in a Bayesian network. A collider (X,Y,Z) is called *unshielded* if X is not adjacent to Z and *shielded* otherwise. Unshielded colliders are very important for Bayesian networks, for they characterize families of networks that cannot be distinguished by observational data alone, as we will see later in this section.

2.3 D-Separation

Definition 2.3.1 (D-separation) *In a DAG $D = (V,E)$, a path π between X and Y is **d-connecting** relative to (condition to) a (possibly empty) set of vertices $\mathbf{Z}$, $\mathbf{Z} \subseteq \mathbf{V} \setminus \{X,Y\}$ if*

1. *Every non-collider on π 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 **d-separated** by $\mathbf{Z}$ if there is no d-connecting path between A and B relative to $\mathbf{Z}$. Otherwise, we say they are **d-connected** given $\mathbf{Z}$. We denote the d-separation of A and B given $\mathbf{Z}$ as $DSep(A;B|\mathbf{Z})$.*

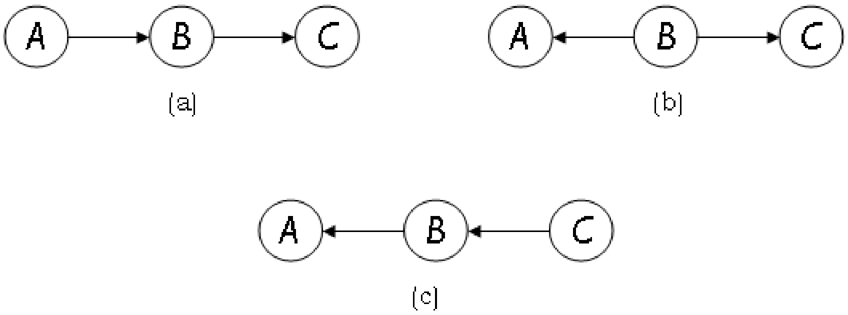


Figure 2.2: The three networks cannot be distinguished statistically.

It is proved [31, 11] that d-separation characterizes all and only the conditional independence relations that follow from the application of the Causal Markov Condition on a faithful causal DAG. D-separation expresses the *global Markov property* for causal DAGs, i.e. an independence holds for the probability distribution if and only if the corresponding d-separation holds in the corresponding causal DAG. Also, notice that in a causal DAG, every non-adjacency corresponds to a conditional independence. This property expresses the *pairwise Markov property* for DAGs.

2.4 Markov Equivalence

While fitting a DAG model for a given dataset is , under the aforementioned assumptions, always possible, a single DAG model is usually not uniquely determined by observational data alone. In the simplest example shown in Figure 2.4, given that A is independent from C given B, all three networks presented are possible. Such models are called *Markov equivalent* and the set of Markov Equivalent models define a *Markov Equivalence Class.* It has been proved [32] that two DAGs are Markov Equivalent if they share the same edges and the same unshielded colliders. In fact, algorithm GES [4] refereed to in the previous section searches for a Markov Equivalence class of DAGs, and not an actual DAG.

2.5 Causal Insufficiency

In the previous sections, we have assumed causal sufficiency. However, it is not often possible that we have measured all the variables necessary to have a causally sufficient system. Moreover, it is often possible that confounding reveals itself. For example, if we observe the joint probability distribution for 4 variables A, B, C, D that are members of the (causally sufficient) system shown

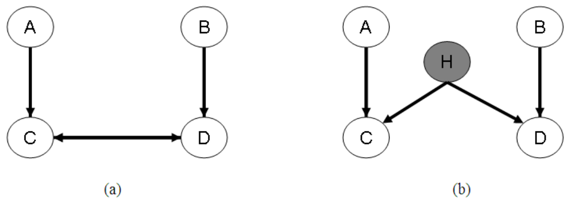


Figure 2.3: In the network in (a), both triples (A, C, D) and (B, D, C) form colliders, revealing causal insufficiency.

in Figure 2.5, then it becomes obvious that there must exist a latent common cause of C and D.

Confounding is very common in practice, and rarely a system should be assumed to meet the causal sufficiency condition. Some of the algorithms described in section 1.2 have been extended to involve latent variables. Some extensions are GES[4] with latent variable post-processing and IC*[18]. In this work, we will deal with latent variables. We have chosen a to use a representation that naturally models the effect of marginalization on DAGs and implicitly includes latent variables. These graphical models, introduced by Richardson and Spirtes [22] are called *Maximal Ancestral Graphs*, and are discussed in detail in the following chapter. FCI [24, 34] is an asymptotically correct algorithm that discovers a Markov Equivalence class of MAGs for a given dataset.

2.6 Factorization and Manipulation

We have discussed the term causal only as a characterization of the edges. However, we have so far not involved any discussion about the actual probabilistic aspects of causal Bayesian networks or manipulation effects. The main effect of $\mathcal{CMC}$ in terms of probabilistic properties is that the probability distribution represented by a DAG $\mathcal{G}$ can be factorized as:

$$P(X_1, X_2, \ldots, X_n) = \prod_i P(X_i | \mathbf{Pa}_{\mathcal{G}}(X_i))$$

limiting the number of parameters significantly. Intervention properties for causal Bayesian Networks have also been studied. Pearl [18] has developed the *do-calculus*, providing rules that allow manipulation-handling. In this work, however, we focus on the qualitative characteristics of causal models, thus, detailed presentation of intervention policies are out of the present scope.

Chapter 3

Maximal Ancestral Graphs

Maximal Ancestral Graphs is a class of graphical independence models introduced in [22] with several desired properties, that render them proper for modeling causal insufficiency. This class of graphical models as described in[22] is designed to cope with both confounding and conditioning. For the purpose of this work, however, only the presence of latent variables is taken under consideration. Basic definitions and concepts are introduced in section 3.1. Maximal Ancestral graphs are connected to independence models described in 3.1 through the notion of M-Separation described in 3.2. In section 3.3 we present some interesting properties of marginalizing independence models and the corresponding graphical models. A single independence model defines a class of Maximal Ancestral Graphs, represented by a Partially Oriented Ancestral Graph described in section 3.4. Finally, in section 3.5 we describe the Fast Causal Inference Algorithm [24, 34], a sound and complete algorithm for recovering a PAG from observational data.

3.1 Basic Definitions

3.1.1 Independence Models

An independence model $\mathcal{J}$ over a set $\mathbf{V}$ of variables is a set of triples $\langle X, Y|\mathbf{Z}\rangle$, where $X, Y, \mathbf{Z}$ are disjoint sets of variables and X, Y are nonempty. Such a triple denotes that X and Y are independent condition to $\mathbf{Z}$, and corresponds to the standard notion of conditional independence in a probability distribution.

A Graph $\mathcal{G}$ is an ordered pair $(\mathbf{V}, \mathbf{E})$ where $\mathbf{V}$ is a set of vertices and $\mathbf{E}$ is a set of edges. Such models are connected to an independence model $\mathcal{J}_C(\mathcal{J})$ through a *separation criterion* C, referred to as *global Markov property*, in the following manner:

$$\langle X, Y|\mathbf{Z}\rangle \in \mathcal{J}_C(\mathcal{G}) \Leftrightarrow X \text{ is separated from } Y \text{ by } \mathbf{Z} \text{ in } \mathcal{G} \text{ by criterion } C$$

We have already described such a criterion, D-Separation [19] for DAGs.

3.1.2 Mixed Graphs

As described in the previous chapter, directed graphs lack to represent confounding, therefore, a wider model must is required.

Definition 3.1.1 *A* mixed graph *is a graph* $\mathcal{G} = (V, E)$ *that can contain directed* $(\rightarrow)$ *and bi-directed* $(\leftrightarrow)$ *edges.*

Given a mixed graph $\mathcal{G} = (V, E)$ and $X, Y \in V$, adjacent in $\mathcal{G}$, X is a *parent* of Y if $X \rightarrow Y \in E$; X is called a *spouse* of Y if $X \leftrightarrow Y \in E$. A vertex cannot be adjacent to itself. Obviously DAGs are mixed graphs(containing only directed edges).

3.1.3 Paths and Ancestors

A path in G is a sequence of distinct vertices $\langle V_0, V_1, \dots, V_n \rangle$ s.t $\forall 0 \leq i < n$, V_i and V_{i+1}, are adjacent in G, and no vertex appears more than once in the sequence.. A path from V_0 to V_n is *directed* if $\forall 0 \leq i < n$, 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 G. $\mathbf{Pa}_{\mathcal{G}}(X), \mathbf{Ch}_{\mathcal{G}}(X), \mathbf{Sp}_{\mathcal{G}}(X), \mathbf{An}_{\mathcal{G}}(X)$ and $\mathbf{De}_{\mathcal{G}}(X)$ are used to denote the set of parents, children, spouses, ancestors and descendants of node X in G, respectively. A *directed cycle* in G occurs when $X \rightarrow Y \in E$ and $Y \in \mathbf{An}_{\mathcal{G}}(X)$. An *almost directed cycle* in G occurs when $X \leftrightarrow Y \in E$ and $Y \in \mathbf{An}_{\mathcal{G}}(X)$.

3.2 M-Separation

The criterion of M-Separation is the graphical criterion consisting the global Markov Property for Maximal Ancestral Graphs.

3.2.1 Ancestral Graphs

The class of mixed graphs is too wide for the purposes of representing DAG models under marginalization. We now present the subclass of ancestral graphs [22].

Definition 3.2.1 *A mixed graph* $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ *is* ***ancestral*** *if for every* $X \in \mathbf{V}$, $X \notin (\mathbf{Sp}_{\mathcal{G}}(X) \cup \mathbf{An}_{\mathcal{G}}(X))$.

This condition states the motivation behind the term "ancestral". In words, it ensures that if X and Y are joined with an edge pointing towards X, X cannot be also an ancestor of Y.‘ The definition can be restated as follows:

Definition 3.2.2 *A mixed graph is* **ancestral** *if the graph does not contain any directed or almost directed cycles.*

Intuitively, this stems from the semantic interpretation of arrowheads in Ancestral Graphs. The following lemma summarizes the semantics of edges in an ancestral graph.

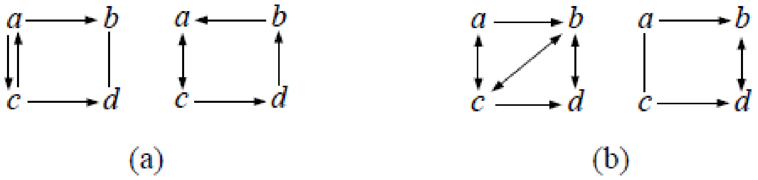


Figure 3.1: Mixed graphs that are not ancestral.

Lemma 3.2.0.1 *[22]. If $\mathcal{G}$ is an ancestral graph, and X, Y are adjacent in $\mathcal{G}$, then:*

1. $X \rightarrow Y \Leftrightarrow X \in \mathbf{An}_{\mathcal{G}}(Y), Y \notin \mathbf{An}_{\mathcal{G}}(X)$
2. $X \leftrightarrow Y \Leftrightarrow X \notin \mathbf{An}_{\mathcal{G}}(Y), Y \notin \mathbf{An}_{\mathcal{G}}(X)$

In a causal framework ,arrowhead denotes "non-ancestry". The definition of an ancestral graph ensures that the graphical model shows no contradicting evidence. No causal feedbacks are allowed. The presence of a directed cycle or an almost directed cycle would denote that a vertex is both an ancestor and a non-ancestor of another vertex, and is therefore rejected. Examples of ancestral and non-ancestral mixed graphs is shown in figure 3.2.1[22].

3.2.2 Maximality

Ancestral Graphs are connected to independence models in a manner similar to the connection of d-separation and DAGs. M-connecting paths are a "natural extension" of d-connecting paths in graphs that can also contain bi-directed edges. On a paths $\langle X_0, X_1, \ldots, X_n \rangle$ a non-endpoint vertex X_i is a *collider* on the path if both X_{i-1} and X_{i+1} have an arrowhead pointing towards X_i. The triple (X_{i-1}, X_i, X_{i+1}) is also said to *form a collider*(obviously on any path). Any non-endpoint vertex is not a collider on the path is said to be a *non-collider* on the path.

Definition 3.2.3 (M-separation) *In a mixed graph $G = (V, E)$, a path π between X and Y is* ***m-connecting*** *relative to (condition to) a (possibly empty) set of vertices $\mathbf{Z}$, $\mathbf{Z} \subseteq \mathbf{V} \setminus \{X, Y\}$ if*

1. *Every non-collider on π 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 $MSep(A; B|\mathbf{Z})$.*

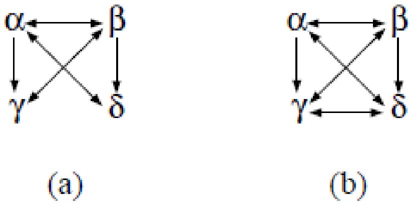


Figure 3.2: (a) An ancestral graph that is not maximal (does not satisfy the pairwise markov property)(b) The maximal ancestral graph corresponding to the same independence model.

M-separation criterion expresses the global Markov property in mixed graphs. The independence model resulting from applying this criterion to a graph $\mathcal{G}$ is denoted $\mathcal{J}_m(\mathcal{G})$. In a DAG, a path is d-connecting if and only if it is m-connecting. However, independence models described by DAGs also satisfy pairwise Markov properties (with respect to the corresponding DAGs), thus, every missing edge corresponds to a conditional independence. This is not the case in ancestral graphs. Figure 3.2.2 [22] depicts an ancestral graph (and the corresponding independence model) that does not satisfy the pairwise Markov property. This motivates the introduction of a new notion; *maximality.*

Definition 3.2.4 *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 $\langle X, Y, \mathbf{Z}\rangle \in \mathcal{J}_m(\mathcal{G})$.*

This definition results in models that also satisfy the pairwise Markov property; hence, in a maximal ancestral graph, every missing edge corresponds to at least one conditional independence in the corresponding independence model. Obviously, a DAG is a maximal ancestral graph. The term maximal refers to the fact that no extra edges may be added in such a graph without changing the independence model, whereas any ancestral graph can be extended to a maximal ancestral graph for the same independence model with the addition of bi-directed edges. An example of such a transformation is shown in Figure 3.2.1(b).

3.3 Marginalization

In the scenario we attempt to deal with, latent variables is an option, therefore the results of marginalization in both independence models and the related graphical models is of great interest. We have previously stated that, under the Causal Markov and Causal Faithfulness Conditions, and in cases of causal sufficiency, an independence model can always be represented by a DAG using the criterion of d-separation. We have also stated that d-separation and m-separation coincide for DAGs. Therefore, assuming any causal data-generating process satisfying $\mathcal{CMC}$ and $\mathcal{CFC}$, a Maximal Ancestral Graph that is directed and acyclic faithfully represents the independence facts stemming from a probability distribution. But what happens when we only observe marginal distributions of the true causal process?

An independence model $\mathcal{J}$ with vertex set $\mathbf{V}$ is a set of triples $\langle X, Y|\mathbf{Z}\rangle$ as mentioned in section 3.1. The result of marginalizing out a set of nodes $\mathbf{L}$, is the subset of triples containing only independencies that do not involve vertices in $\mathbf{L}$:

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

The corresponding transformation for graphical models is defined as follows:

Definition 3.3.1 *If $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ is an ancestral graph, graph $\mathcal{G}[_{\mathbf{L}}$ 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*

$$\left\{\begin{array}{l} X \notin \mathbf{An}_{\mathcal{G}}(Y); Y \notin \mathbf{An}_{\mathcal{G}}(X) \\ X \in \mathbf{An}_{\mathcal{G}}(Y); Y \notin \mathbf{An}_{\mathcal{G}}(X) \\ X \notin \mathbf{An}_{\mathcal{G}}(Y); Y \in \mathbf{An}_{\mathcal{G}}(X) \end{array}\right\} \text{ then } \left\{\begin{array}{c} X \leftrightarrow Y \\ X \rightarrow Y \\ X \leftarrow Y \end{array}\right\} \text{ in } \mathcal{G}[_{\mathbf{L}}$$

This graphical transformation maps an ancestral graph after marginalizing out a set of vertices to what is proved to be, a maximal ancestral graph. Thus, maximal ancestral graphs are closed under marginalization. Closure of the set of maximal ancestral graphs under marginalization is a key point for the development of our algorithm. This result, along with some more interesting properties, have been have been proved in [22].. In the rest of this chapter some of these properties which comprise the theoretical foundation of our algorithm.

3.3.1 Inducing Paths

Inducing Paths are special paths introduced in [32], closely relevant to the notions of maximality and marginalization.

Definition 3.3.2 *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 $\mathbf{L} = \varnothing$, π is called a primitive inducing path.*

Intuitively, an inducing path w.r.t. $\mathbf{L}$ is a path that may not be blocked condition to any subset of $\mathbf{V} \setminus \mathbf{L}$. This means that, when marginalizing out a set of vertices $\mathbf{L}$, for every pair of vertices $X, Y \in \mathbf{V} \setminus \mathbf{L}$, if there exists in $\mathcal{G}$ an inducing path w.r.t. $\mathbf{L}$, X and Y cannot be m-separated in $\mathcal{G}[_{\mathbf{L}}$. The following theorem [22] summarizes the relation between m-separation and inducing paths.

An edge between $X, Y \in \mathbf{V} \setminus \mathbf{L}$ is considered (trivially) an inducing path w.r.t. $\mathbf{L}$.

Theorem 3.3.1 *If $\mathcal{G}$ is an ancestral graph, with vertex set $\mathbf{V} = \mathbf{O} \dot{\cup} \mathbf{L}$, then the following four conditions are equivalent.*

1. *There is an edge between X and Y in $\mathcal{G}[_{\mathbf{L}}$*
2. *There is an inducing path between X and Y w.r.t. $\mathbf{L}$ in $\mathcal{G}$*
3. $\forall \mathbf{Z}, \mathbf{Z} \subseteq \mathbf{V} \setminus (\mathbf{L} \cup \{X, Y\}), \langle X, Y | \mathbf{Z} \rangle \notin \mathcal{J}_m(\mathcal{G})$.
4. $\forall \mathbf{Z}, \mathbf{Z} \subseteq \mathbf{V} \setminus (\mathbf{L} \cup \{X, Y\}), \langle X, Y | \mathbf{Z} \rangle \notin \mathcal{J}_m(\mathcal{J}[_{\mathbf{L}})$

A result of this theorem is the following proposition [34], that describes the connection between primitive inducing paths and maximality:

Proposition 3.3.1 *An ancestral graph is maximal if and only if there is no primitive inducing path between any two non-adjacent vertices.*

3.3.2 Marginalizing Ancestral Graphs

This section summarizes the two most appealing features of maximal ancestral graphs: Equivalence of independence models and graphical models (through the global Markov property) under marginalization, and closure. These results have been proved in [22], and comprise the two most appealing features of maximal ancestral graphs when it comes to representing marginal distributions. The following theorem connects marginalization for independence models and the corresponding ancestral graphs.

Theorem 3.3.2 *If $\mathcal{G}$ is an ancestral graph over $\mathbf{V}$ and $\mathbf{L} \subset \mathbf{V}$, then*

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

The following corollary states closure under marginalization for the set of maximal ancestral graphs:

Corollary 3.3.2.1 *If $\mathcal{G}$ is an arbitrary ancestral graph with vertex set $\mathbf{V} = \mathbf{O} \dot{\cup} \mathbf{L}$, $\mathcal{G}[_{\mathbf{L}}$ is a maximal ancestral graph.*

The above two results render maximal ancestral graphs a powerful tool for modeling marginal distribution. In the scenario we attempt to deal with, the distributed independence models are considered marginals of the independence model describing the causal relations over the complete set of variables measured, therefore closure under marginalization is a key property.

3.4 Markov Equivalence

An independence model does not uniquely define a maximal ancestral graphs. The same correlational pattern may be shared by several different MAGs, all possible data- generating structures for our observed data. Like in DAGs, statistical indistinguishability also holds for MAGs. An example of two MAGs

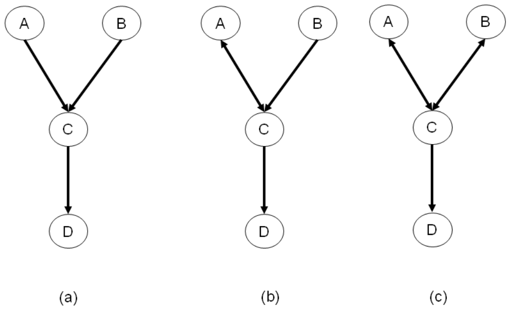


Figure 3.3: Markov Equivalent Mags that cannot be statistically distinguished.

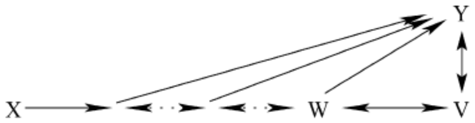


Figure 3.4: A discriminating path for V from X to Y. If V is not a member of any set separating X and Y, V is a collider on the path (triple (W, V, Y) is a collider).

sharing the same m-separation structure but representing different causal information is shown in Figure 3.4. MAGs over the same vertex set that share the same correlational characteristics define a Markov equivalence class.

Definition 3.4.1 *Two MAGs $\mathcal{G}_1, \mathcal{G}_2$ over the set of vertices are said to be* Markov equivalent *if for any three disjoint sets $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$, $\mathbf{X}$ and $\mathbf{Y}$ are m-separated by $\mathbf{Z}$ in $\mathcal{G}_1$ if and only if $\mathbf{X}$ and $\mathbf{Y}$ are m-separated by $\mathbf{Z}$ in $\mathcal{G}_2$.*

This definition states that members of a Markov equivalence class entail the same conditional independencies. But what are the common characteristics shared by Markov Equivalent MAGs? In chapter 1, we saw that two DAGs are Markov Equivalent if and only if they share the same adjacencies and unshielded colliders [32]. To characterize Markov equivalence for MAGs, however, some shielded colliders have to be taken under consideration. These shielded colliders are defined by a special kind of paths, *discriminating paths.*

Definition 3.4.2 *A path $p = \langle X, \ldots, W, V, Y \rangle$ is called a* discriminating *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.*

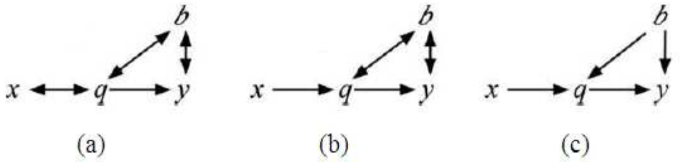


Figure 3.5: All three MAGs have the same adjacencies and unshielded colliders, but (a) and (c) are not Markov equivalent. $\langle x, q, b, y \rangle$ forms a discriminating path for b in every MAG.

The structure of a discriminating path is depicted in Figure 3.4. Discriminating paths can also appear in DAGs, but, due to lack of bi-directed edges, they always discriminate non-colliders and therefore play no role in Markov equivalence. In MAGs, however, when a discriminating path is present, it "discriminates" a vertex on the path, in the sense that the vertex is either a collider or a non-collider for a specific independence model. A detailed example is provided in Figure 3.4[1]. Therefore, two MAGs over the same vertex that share the same discriminating path can only be Markov equivalent if the vertex being "discriminated" is a collider in either both or neither of the graphs. In other words, discriminating paths are similar to unshielded triples: The discriminated vertex is either a member of all the sets separating the path endpoints (therefore a collider), or a member of all the sets that m-separate the path endpoints (therefore a non-collider). There are many characterizations for Markov equivalence in ancestral graphs[1]. In this work, we shall only use the following, proved by Spirtes and Richardson[26].

Proposition 3.4.1 *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.*

Given a dataset of observed variables, we can identify the Markov equivalence class of MAGs deriving from the associative structure of our data, using the Fast Causal inference Algorithm described in section 3.5. We will denote the set of MAGs that are Markov equivalent to a MAG $\mathcal{G}$ by $[\mathcal{G}]$, following the notation used in [34].

3.4.1 Partially Oriented Ancestral Graphs

Markov equivalent MAGs share a several common characteristics. Like P-DAGs for DAGs, invariant characteristics of a Markov equivalence class can be summarized in a graph that shares all the invariant characteristics of $[\mathcal{G}]$.

Definition 3.4.3 *(Partial Ancestral Graph) Let $[\mathcal{G}]$ be the Markov equivalent class for a MAG $\mathcal{G}$. A* Partial Ancestral Graph *is a graph $\mathcal{P}$ containing (up to) three kinds of endpoints: arrowhead ($>$), tail ($-$), and circle ($\circ$) , with the following properties:*

1. *$\mathcal{P}$ has the same adjacencies as any member of the equivalence class.*
2. *Every non-circle endpoint in $\mathcal{P}$ 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. FCI algorithm as presented in [25] is a sound algorithm that, given an oracle of conditional independence, provides a PAG for the Markov equivalence class of the true causal MAG over a set of variables.

Definition 3.4.4 *(Maximally Informative PAG) If $\mathcal{P}$ is a PAG corresponding to a Markov equivalence class $[\mathcal{G}]$, and every circle in $\mathcal{P}$ corresponds to a variant mark in $[\mathcal{G}]$, $\mathcal{P}$ is the* maximally informative PAG *for $[\mathcal{G}]$.*

Zhang in [34] provided an extended version of FCI, which is provably sound and complete, and, therefore, given an oracle of conditional independence, returns the maximally informative PAG over the set of variables observed in a single dataset. We shall henceforth assume that the PAGs used have been derived from the extended FCI algorithm (as described in the following section) or some other complete process, and by the term PAG we shall only refer to maximally informative partially oriented ancestral graphs.

The PAG for the Markov equivalence class of MAGs in Figure 3.4 is shown in Figure 3.4.1. Notice that circle endpoints in a PAG cannot be oriented arbitrarily to form a possible data-generating MAG. Fitting MAG or DAG models when the PAG of their Markov equivalence class is known can be found in [33].

3.5 Algorithm *Fast Causal Inference*

FCI [25, 34] is an asymptotically correct algorithm that discovers the PAG "representing" the true causal structure (MAG) over a set of variables in the presence of latent variables and selection bias. For the purposes of this work, selection bias is not taken under consideration. Therefore, we describe a modified version of the algorithm limited to latent variable modeling. Before presenting the algorithm, we introduce a few more notions from necessary.

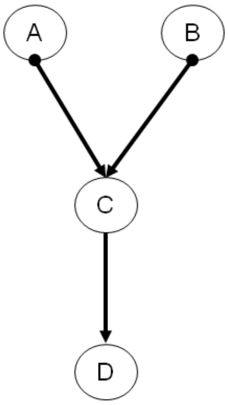


Figure 3.6: The Partially Oriented Ancestral graph representing the Markov equivalence class of MAGs in Figure 3.4.

Definition 3.5.1 *(uncovered path) In a partially oriented mixed graph, a path $p = \langle V_0, \dots, V_n \rangle$ is said to be* uncovered *if for every $1 \leq i \leq i-1$, V_{i-1} and V_{i+1} are not adjacent.*

In words, in an uncovered path, every consecutive triple is unshielded.

Definition 3.5.2 *In a partially oriented mixed graph, a path $p = \langle V_0, \dots, V_n \rangle$ is said to be* potentially directed *if for every $1 \leq i \leq i-1$, the edge between V_i and V_{i+1} is not into V_i or out of V_{i+1}.*

Thus, the term potentially directed path is used to denote a path that could be directed, i.e. there exists a configuration of circle endpoints rendering the path directed. We now describe the version of Fast Causal Inference Algorithm (FCI)[1]. as implemented for this thesis. Asterisk is used in orientation rules as a meta symbol; that is, to denote any possible configuration of an endpoint (arrowhead, tail or circle). The details on how the adjacency search is performed are omitted, however, the implementation follows the procedure described in [24], which ensures that (given an oracle of conditional independence) if there exists a separating set for a pair of vertices, it will be discovered.

The orientation rules 1-3 and 8-10 are graphically illustrated in Figures 3.5 and 3.5(adopted from [34]), respectively. The rule numbering also follows that in [34]. Rules 5-7 in [34] refer to cases of selection bias (depicted in PAGs as undirected edges). Since we do not deal with this issue in the present work, we have omitted these rules. The output of Algorithm 1 is a maximally informative partially oriented ancestral graph representing the Markov equivalence class of

[1] Actually, the algorithm that includes rule 8-10 and an additional set of rules concerning conditioning is referred to as AFCI in [34]In this work, we call FCI algorithm 1

Algorithm 1: FCI algorithm

```
Input: Dataset $\mathcal{D}$ over variables $\mathbf{O}$
Result: PAG $\mathcal{P}$ over variables $\mathbf{O}$
$\mathcal{P} \leftarrow$ Complete graph over $\mathbf{O}$;
For every pair of variables $\alpha, Y$ if $\exists \mathbf{S} \subset \mathbf{O} \setminus \{\alpha, \beta\}$ s.t. $\alpha \perp \beta | \mathbf{S}$ then
    Remove edge $(\alpha, \beta)$ from $\mathcal{P}$;
    Record $\mathbf{S}$ as $SepSet(\alpha\beta)$;
end
/* $\mathcal{R}0$: orient unshielded colliders */
for each unshielded triple $(\alpha, \gamma, \beta)$ in $\mathcal{P}$ do
    orient it as a collider $\alpha * \rightarrow \gamma \leftarrow * \beta$ if and only if $\beta \in SepSet(\alpha\beta)$
end
. Apply the following orientation rules until none of them applies:
/* $\mathcal{R}1$:away from collider */
if $\alpha * \rightarrow \beta \circ - * \gamma$ and $\alpha$ is not adjacent to $\gamma$ then
    Orient the triple as $\alpha * \rightarrow \beta \rightarrow \gamma$
end
/* $\mathcal{R}2$: away from ancestor */
if $\alpha \rightarrow \beta \rightarrow \gamma$ or $\alpha * \rightarrow \beta \rightarrow \gamma$ and $\alpha * - \circ \gamma$ then
    orient $\alpha * - \circ \gamma$ as $\alpha * \rightarrow \gamma$
end
/* $\mathcal{R}3$:double triangle */
if $\alpha * \rightarrow \beta \leftarrow * \gamma$, $\alpha * - \circ \theta \circ - * \gamma$, $\alpha$ and $\gamma$ are not adjacent and $\theta * - \circ \beta$
then
    orient $\theta * - \circ \beta$ as $\theta * \rightarrow \beta$
end
/* $\mathcal{R}4$ : discriminating path */
if $p = \langle \theta, \ldots, \alpha, \beta, \gamma \rangle$ is a discriminating path between $\theta$ and $\gamma$ for $\beta$, and
$\beta \circ - * \gamma$ then
    if $\beta \in SepSet(\theta\gamma)$ then
        orient $\beta \circ - * \gamma$ as $\beta \rightarrow \gamma$
    end
    else
        orient triple $(\alpha, \beta, \gamma)$ as $\alpha \leftrightarrow \beta \leftrightarrow \gamma$
    end
end
/* $\mathcal{R}8$ */
if $\alpha \rightarrow \beta \rightarrow \gamma$ or $\alpha - \circ \beta \rightarrow \gamma$, and $\alpha \circ \rightarrow \gamma$ then
    orient $\alpha \circ \rightarrow \gamma$ as $\alpha \rightarrow \gamma$
end
/* $\mathcal{R}9$ */
if $\alpha \circ \rightarrow \beta$, and $p = \langle \alpha, \beta, \theta, \ldots, \gamma \rangle$ is an uncovered p.d. path from $\alpha$ to $\gamma$
such that $\beta$ and $\gamma$ are not adjacent then
    orient $\alpha \circ \rightarrow \gamma$ as $\alpha \rightarrow \gamma$
end
/* $\mathcal{R}10$ */
Suppose $\alpha \circ \rightarrow \gamma$, $\beta \rightarrow \gamma \leftarrow \theta$, $p_1$ is an uncovered p.d. path from $\alpha$ to $\beta$, $p_2$
is an uncovered p.d. path from $\alpha$ to $\theta$. Let $\mu$ be the vertex adjacent to $\alpha$
on $p_1$ ($\mu$ could be $\beta$), and $\omega$ be the vertex adjacent to $\alpha$ on $p_2$ ($\omega$ could
be $\theta$).
if $\mu$ and $\omega$ are distinct, and not adjacent then
    orient $\alpha \circ \rightarrow \gamma$ as $\alpha \rightarrow \gamma$.
end
return $\mathcal{P}$;
```

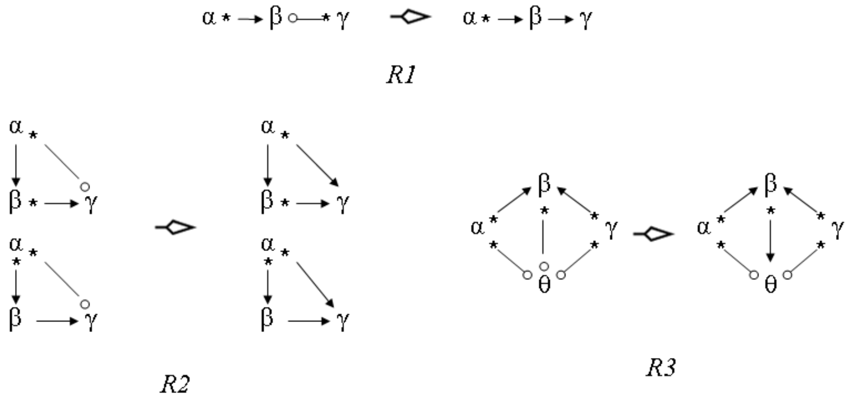


Figure 3.7: FCI Orientation Rules $\mathcal{R}1 - \mathcal{R}3$. When the conditions of the corresponding rules are met, the pattern on the left is substituted by the pattern on the right. Essentially the same rules are used when learning causal DAGs.

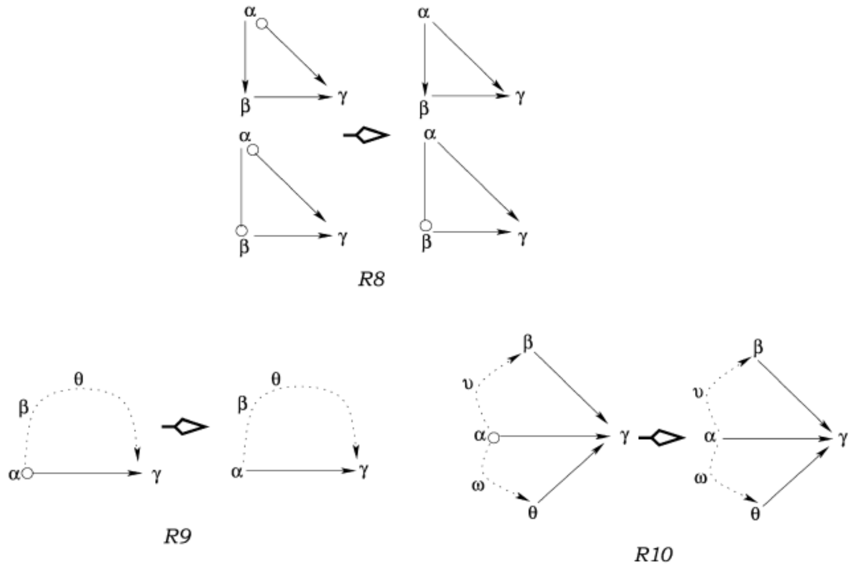


Figure 3.8: FCI Orientation Rules $\mathcal{R}8$ – $\mathcal{R}10$. When the conditions of the corresponding rules are met, the pattern on the left is substituted by the pattern on the right. These rules are used to turn partially directed edges (∘ →) into directed ones (→).

true MAG $\mathcal{M}$ that generates the given dataset, when given correct information on conditional independencies.

For the purpose of this work, we have chosen not to deal with flaws of statistical information based on observational data, and from now on we shall assume that we are given an oracle of conditional independence.

Chapter 4

A novel constraint-based algorithm for inferring causal structure from overlapping variable sets

In the previous chapter, we presented a graphical model suitable for modeling marginal distributions [22], and a sound and complete algorithm that discovers the invariant characteristics graphical models representing an independence model, based on observational data. In this chapter, we will make use of the desirable properties of causal relationships to combine causal structures over overlapping variable sets obtained from different observations. The main purpose of this algorithm is to utilize the available datasets concerning similar but not identical variable sets, use FCI algorithm to obtain the PAGs over these variable sets, and combine these causal structures into a constraint satisfaction problem whose solutions are the possible data-generating causal structure. However, the distributed causal structures can be obtained by any other sound and complete method(e.g. GES with latent variable preprocessing steps, domain knowledge).

4.1 Problem Definition

We assume that we are given access to a set of independence models $\{\mathcal{J}_i\}_{i=1}^N$ over corresponding subsets of variables $\mathbf{O}_i$. We define the problem of identifying a MAG consistent with all $\mathcal{J}_i$ where we use the notation $\overline{\mathbf{O}_i} \equiv \mathbf{O} \setminus \mathbf{O}_i$, where $\mathbf{O} = \bigcup_{i=1}^N \mathbf{O}_i$.

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

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

The approach used to solve this problem assumes is that there exists a single underlying causal mechanism ruling the complete set of observed variables. of course, this mechanism could involve additional variables, which we do not observe. Thus, we assume that a single mechanism over variables $\mathbf{O}\cup\mathbf{L}$ generates the data. $\mathbf{L}$ stands for the set latent variables that are involved in the true causal procedure, but have not been measured in any dataset, and therefore $\mathbf{O} \cap \mathbf{L} = \varnothing$. There exists a single independence model $\mathcal{J}$ over $\mathbf{O} \cup \mathbf{L}$. Under the Causal Markov and the Causal Faithfulness Conditions, this mechanism can be represented by a DAG $\mathcal{D}$, and the associative properties can be obtained from the graph through the criterion of m-separation [1]. So, in principle, the observed data correspond to the marginal distributions $\mathcal{J}[_{\overline{\mathbf{O_i}}\cup\mathbf{L}}$,thus, by theorem 3.3.2, PAGs $\mathcal{P}_i$ over $\mathbf{O}_i$ represent the Markov equivalence classes of MAGs $\mathcal{G}[_{\overline{\mathbf{O_i}}\cup\mathbf{L}}$. Therefore, the problem can be recast as:

Problem 2 (Find Consistent MAG) *Given Partially Oriented Ancestral Graphs* $\{\mathcal{P}_i\}_{i=1}^N$ *representing Markov equivalence classes of MAGs* $[\mathcal{G}]$ *over subsets of variables* $\mathbf{O}_i$*, induce a MAG* $\mathcal{M}$ *s.t., for all i*

$$\mathcal{M}[_{\overline{\mathbf{O}_i}} \in [\mathcal{G}_i]$$

Thus, PAGs that have occurred from any provably sound procedure can be used as input for the algorithm we present. The idea is, that the distributively learned structures must be combined in a way that preserves the m-separation and m-connection properties that occur in $\mathcal{J}[_{\overline{\mathbf{O_i}}\cup\mathbf{L}}$. These properties form the constraints of our problem. Before analyzing the generation of the constraints, we must introduce some basic notions.

4.2 Preliminaries

We hereby introduce some notions that constitute the base of our problems. In order to transform a set of input graphical models into a set of binary constraints, we have developed a "language" describing mainly graph path properties. The main notions of this language are hereby presented. We will henceforth refer to these notions as *predicates*, even though they are not defined in e strict formal language framework.

Definition 4.2.1 *(Main Predicates) For a mixed graph* $\mathcal{G} = (\mathbf{V}, \mathbf{E})$*, we define the following predicates:*

1. *edge*(X, Y) *is true when* X *and* Y *are adjacent (*$X, Y \in \mathbf{V}$ *and* $(X, Y) \in \mathbf{E}$*).*

[1] As mentioned in section 3.2, DAGs are maximal ancestral graphs for which the criterion of d-separation coincides with that of m-separation

2. *$arrowhead(X, Y)$ is true when X is into Y. $X, Y \in \mathbf{V}$ and $(X{*}\!\rightarrow Y) \in \mathbf{E}$*

These are the main variables used for the generation of the constraints. Graph and path properties described below expressed in terms of these two initial predicates using the rules in 4.2.3

Definition 4.2.2 *(Additional predicates) For a mixed graph $(G) = (\mathbf{V}, \mathbf{E})$ and a path $p = \langle X_0, X_1 \ldots, X_n \rangle$ we define the following predicates:*

1. *$collider(X, Y, W)$ is true when triple (X, Y, W) forms a collider.*
2. *$ancestral(p)$ is true when p is ancestral (directed).*
3. *$ancestor(X, \mathbf{Y})$ is true when X is an ancestor of some member of $\mathbf{Y}$.*
4. *$inducing(p, \mathbf{Z})$, $\mathbf{Z} \subset \mathbf{V}$, is true when p is inducing w.r.t $\mathbf{Z}$.*
5. *m-connecting$(p, \mathbf{Z})$, $\mathbf{Z} \subset \mathbf{V}$, is true when p is inducing condition to $\mathbf{Z}$.*

These additional predicates described above can be expressed in terms of the main predicates as follows:

Definition 4.2.3

$$\begin{aligned}
collider(X, Y, W) &\leftrightarrow edge(X, Y) \wedge edge(Y, W) \wedge arrowhead(X, Y) \wedge arrowhead(Y, W))\\
ancestral(p) &\leftrightarrow \bigwedge_{1 \le i \le n} \big(edge(X_{i-1}, X_i) \wedge arrowhead(X_{i-1}, X_i) \wedge \neg arrowhead(X_i, X_{i-1})\big)\\
ancestor(X, \mathbf{Y}) &\leftrightarrow \exists p\big(X_0 = X \wedge X_n \in Y \wedge ancestral(p)\big)\\
inducing(p, \mathbf{Z}) &\leftrightarrow \bigwedge_{1 \le i \le n-1} \Big(edge(X_{i-1}, X_i) \wedge edge(X_i, X_{i+1}) \wedge\\
&\quad \big(X_i \in \mathbf{Z} \rightarrow \neg collider(X_{i-1}, X_i, X_{i+1}) \vee ancestor(X_i, \{X_0, X_n\})\big) \wedge\\
&\quad \big(X_i \notin \mathbf{Z} \rightarrow collider(X_{i-1}, X_i, X_{i+1}) \wedge ancestor(X_i, \{X_0, X_n\})\big)\Big)\\
m\text{-}connecting(p, \mathbf{Z}) &\leftrightarrow \bigwedge_{1 \le i \le n-1} \Big(edge(X_{i-1}, X_i) \wedge edge(X_i, X_{i+1}) \wedge\\
&\quad \big(X_i \in \mathbf{Z} \rightarrow collider(X_{i-1}, X_i, X_{i+1})\big) \wedge\\
&\quad \big(X_i \notin \mathbf{Z} \rightarrow \neg collider(X_{i-1}, X_i, X_{i+1}) \vee ancestor(X_i, \mathbf{Z}\})\big)\Big)
\end{aligned}$$

These rules stem directly from the corresponding definitions. A triple is (forms) a *collider* when both endpoint vertices point towards the middle vertex. An *ancestral path* is a directed path, hence, every participating edge is directed (towards the same direction). A vertex X is an *ancestor* of a set of vertices $\mathbf{Y}$ if it is an ancestor of some member of the set, i.e. if there exists an ancestral path from X to some member of $\mathbf{Y}$. A path is *inducing* w.r.t. a set of vertices $\mathbf{Z}$ collider on the path is an ancestor of one of the endpoints and every non-collider is in $\mathbf{Z}$. Thus, if a vertex on the path is in $\mathbf{Z}$, the vertex may be a collider and ancestor of one of the endpoints or a non-collider. Accordingly, if a vertex on the path is not a member of $\mathbf{Z}$, the vertex must be a collider and an ancestor one of the endpoints.

Similarly, an m-connecting path condition to $\mathbf{Z}$ is a path on which every non-collider is not in $\mathbf{Z}$ and every collider is an ancestor of some member of $\mathbf{Z}$. Therefore, if a vertex on the path is in $\mathbf{Z}$, the vertex must be a collider, otherwise the path would be blocked. Accordingly, if a vertex on the path is not in $\mathbf{Z}$, then the vertex can be a collider or a non - collider. However, if the vertex is a collider it must also be an ancestor of some member of $\mathbf{Z}$.

Notice that, in these two rules, one of two implications ($\rightarrow$) is "triggered" for every node. If the node is in $\mathbf{Z}$, only the first of the implications needs to be examined. the chosen formulations serves the following purpose: Once the problem is instantiated, the elements of the language are denoted explicitly and therefore only one of the implications is included in the SAT instance.

4.3 Algorithm *Find Consistent MAG*

What we are looking for is a model over the union of observed variables that justifies the marginal distributions observed. In chapter 3, we analyzed how the notions of conditional independence and dependence are expressed in terms of graphical criteria. The algorithmic approach we hereby follow utilizes this correspondence to combine the independence constraints imposed by distributively gathered data. The transformation lies on the definition of m-separation and theorem 3.3.1.

In particular, suppose that in some marginal independence model$\mathcal{J}_i$ of $\mathcal{J}$, we observe a conditional independence $\langle X, Y|\mathbf{Z}\rangle$, or, respectively, in some PAG $\mathcal{P}_i$ we have found a pair of vertices (X, Y) that are not adjacent, and the recorded separating set is $\mathbf{Z}$[2]. Naturally, the pair of nodes cannot be adjacent in any MAG claiming to be a data-generating process, since the conditional independence $\langle X, Y|\mathbf{Z}\rangle$ must hold in $\mathcal{J}$.In addition, there must be no m-connecting path from X to Y condition to $\mathbf{Z}$. These conditions cover the conditional independencies that are entailed by the marginal independence model $\mathcal{J}_i$.

The conditional dependencies entailed by $\{\mathcal{J}_i\}$, are not so trivial to impose. The presence of an edge (X, Y) in a PAG $\mathcal{P}_i$ denotes that there is no subset of $\mathbf{O}_i$ that renders X and Y independent. However, the presence of this edge may be the result of not simultaneously observing the variables that are necessary render the two vertices independent. However, the presence of every edge in every $\mathcal{P}_i$ must be explicable by the presence of an inducing path w.r.t. $\overline{\mathbf{O}_i}$.

In summary, for any possible data-generating MAG $\mathcal{M}$ over $\mathbf{O}$, the following two conditions must hold:

1. $\forall X, Y$, if X is not adjacent to Y in some $\mathcal{P}_i$, there is no m-connecting path from X to Y condition to $SepSet_i(X, Y)$ in $\mathcal{M}$.

[2] The implemented version of this algorithm includes FCI, and for parsimonious purposes, separating sets are cached and available for further use. However, the algorithm works for any input PAG with annotated separating sets, which can obviously be discovered from any maximally informative PAG.

2. $\forall X, Y$ s. t. X is adjacent to Y in some $\mathcal{P}_i$, there exists an inducing path from X to Y w.r.t. $\overline{\mathbf{O}_i}$ in $\mathcal{M}$

In terms of the predicates described in definition 4.2.2, the aforementioned properties can be expressed as follows:

Proposition 4.3.1 *Given a set of PAGs $\mathcal{P}_i$ representing a set of marginal distributions $\mathcal{J}_i$, if $\mathcal{M}$ is the data -generating MAG, the following must hold in $\mathcal{M}$:*

1. *$\forall X, Y$, if X is not adjacent to Y in some $\mathcal{P}_i$, $(\forall p = \langle X, \dots, Y \rangle) \neg$ m-connecting$(p, SepSet_i(X, Y))$.*

2. *$\forall X, Y$, if X is adjacent to Y in some $\mathcal{P}_i$, $\exists p = \langle X, \dots, Y \rangle)$inducing$(p, \overline{\mathbf{O}_i})$.*

Proof 1) By the definition of m-separation. 2) By theorem 3.3.1, if there is an edge between X and Y in $\mathcal{P}_i$, there exists an inducing path between X and Y w.r.t. $\overline{\mathbf{O}_i}$ in $\mathcal{M}$.

FCM (Find Consistent MAG) is an algorithm that answers problem 2. It takes as input a set of PAGs $\{\mathcal{P}_i\}$ representing $[\mathcal{G}]$ and returns a MAG $\mathcal{K}$ for which $\mathcal{M}[_{\overline{\mathbf{O}_i}}) \in \mathcal{G}_i$.

Algorithm 2: Find Consistent MAG (FCM)

```
Input: PAGs $\{\mathcal{P}_i\}_{i=1}^N$ over variables $\{\mathbf{O}_i\}_{i=1}^N$
Input: Causal Query $\Phi_q$
Result: MAG $\mathcal{K}$
$\mathcal{K} \leftarrow$InitializeGraph $(\{\mathcal{P}_i\}_{i=1}^N)$;
$\Phi_c \leftarrow CNF(\Phi_q)$;
$\Phi_c \leftarrow$GenerateConstraints $(\{\mathcal{P}_i\}_{i=1}^N, \mathcal{K})$;
repeat
    $L \leftarrow$ solveSAT $(\Phi_c)$;
    if $L = \varnothing$ then
        return $\varnothing$
    end
    $\mathcal{K} \leftarrow$ makeChanges $(\mathcal{K}, L)$;
    for each (almost) directed cycle in $\mathcal{K}$ do
        add constraints to $\Phi_c$ preventing cycle
    end
until $\mathcal{K}$ has no (almost) directed cycles ;
return $\mathcal{K}$;
```

4.3.1 Initialization

At first, the input PAGs $\{\mathcal{P}_i\}$ are combined trivially, as described in function 3.

```
Function InitializeGraph($\{\mathcal{P}_i\}_{i=1}^N$)
1 $\mathcal{K} \leftarrow$ complete unoriented graph over $\mathbf{O} = \bigcup_i \mathbf{O}_i$ ;
2 Transfer non-adjacencies and orientations from all $\mathcal{P}_i$ to $\mathcal{K}$ ;
3 Mark all edges as uncertain;
4 return $\mathcal{K}$
```

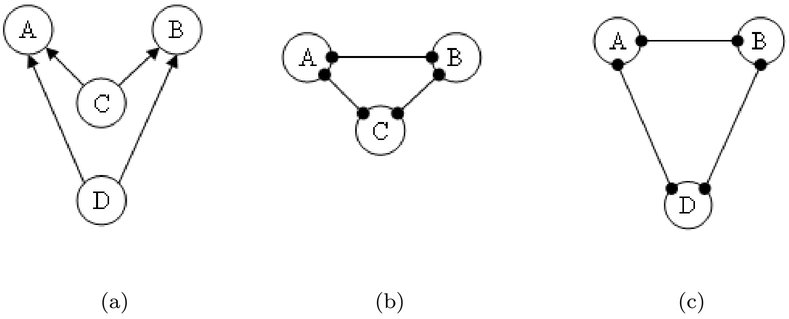


Figure 4.1: An example of edges that appear in sub-structures, but not in the actual structure. (a) The underlying causal structure over $\{A, B, C, D\}$.(b) The causal structure over observed variables $\{A, B, C\}$. (c) The causal structure over observed variables $\{A, B, D\}$. An edge between A and B is present in both sub-PAGs, as a result of not observing C and D simultaneously in any dataset.

We will call the graph resulting from function 3 *initial graph.* This graphical model may contain edges that exist or not in $\mathcal{M}_T$, but if an edge is absent from initial graph, the edge is absent from $\mathcal{M}_T$. The initial graphs also has some oriented edges, which are sound in the following manner: If the edge exists in $\mathcal{M}_T$, it has an orientation permitted by initial graph. Thus, there is no tail(resp. arrowhead) on initial graph that is an arrowhead(resp. tail) on $\mathcal{M}_T$.

All edges of the initial graph the graph may or may not be present on $\mathcal{M}_T$. Edges connecting vertices we have never observed together are included, and are of course uncertain, since we have no information on their existence. Edges that have been observed in all sub-structures that include its endpoints are also marked as uncertain, since their presence could be the result of not observing simultaneously the entire set of variables required to render them independent. An example is shown in Figure 4.3.1, where edge (A, B) is present in both PAGs, but not in $\mathcal{M}_T$.

Thus, all edges in initial graph are marked as uncertain. However, for edges that have been observed in some $\mathcal{P}_i$, regardless of whether they exist in initial graph or not, their presence must be explained. Uncertain edges and uncertain orientations (circle endpoints) will be the main variables of our problem, and the only variables needed to form the boolean constraints. For example, the initial graph that would occur from trivially combining PAGS in Figure 4.3.1

(b) and (c), which is a complete graph over $\{A, B, C, D\}$ with circles at every endpoint, all edges and all endpoints are includes in the *main variables set.* However, some additional variables will be used to facilitate the conversion of the SAT instance in conjunctive normal form.

4.3.2 Generation of the constraints

The algorithm proceeds by including the constraints that transform the problem instance into as boolean formula. Algorithm 4 describes this procedure.

```
Algorithm 4: GenerateConstraints
   Input: PAGs $\{\mathcal{P}_i\}_{i=1}^N$ over $\{\mathbf{O}_i\}_{i=1}^N$, graph $\mathcal{K}$ over $\bigcup_i^N \mathbf{O_i}$
   Result: boolean formula $\Phi_c$ in CNF
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 $\Phi_c$ ;
```

At lines 1 - 4 of Algorithm 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. Algorithm 11 describes the generation of these constraints. The predicates referred to in this algorithm are expressed , according to he rules described in 4.2.3, in terms of the main variables and only. In addition, the inducing paths attempting to substitute for an edge are required to be non-primitive (imposed by $\neg inducing(path, \varnothing)$).[3] This way the resulting graph maintains the maximality property, at least for edges that have been encountered at least in one sub-PAG. 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. $\mathbf{L}$. The procedure is implemented using an extended version of Algorithm 2.2 in [15] for finding d-separations in a DAG. The modifications concern two aspects: First, the paths explored are recorded; second, the algorithm referred to uses an initial preprocessing step that generates an ancestry matrix for a DAG. Using a similar procedure, we generate a matrix of "possible ancestry", while caching the possible ancestral paths.

At lines 5 - 8, edges that have been eliminated from some $\mathcal{P}_i$ are iterated.

[3] By definition of inducing path, every path wick is inducing relative to $\varnothing$ is also primitive relative to any set $\mathbf{L}$, therefore we only need to look for paths that are inducing relative to $\overline{\mathbf{O}_i}$ for every i necessary.

```
Function AdjacencyConstraints($X, Y, \mathbf{L}, \mathcal{K}$)
$\Phi_c \leftarrow \varnothing$;
paths$\leftarrow$possibleInducingPaths($X, Y, \mathbf{L}, \mathcal{K}$);
for each path $\in$ paths do
    $\Phi_c \leftarrow \Phi_c \vee inducing(path, \mathbf{L})$;
end
if $X, Y$ are adjacent in $\mathcal{K}$ then
    $\Phi_c \leftarrow \Phi_c \vee edge(X, Y)$;
    for each path $\in$ paths do
        $\Phi_c \leftarrow \Phi_c \wedge (edge(X, Y) \vee \neg inducing(path, \varnothing))$;
    end
end
return CNF($\Phi_c$)
```

A missing edge (X, Y) corresponds to at least one conditional independency $\langle X, Y | SepSet_i(X, Y)\rangle$ found by FCI when inducing $\mathcal{P}_i$. The separating sets $SepSet_i(X, Y)$ are cached during execution of the FCI algorithm so they are not rediscovered. For every missing edge (X, Y) in $\mathcal{P}_i$, a set of constraints is added to the formula requiring that no m-connecting path exists in $\mathcal{K}$ between (X, Y) condition on $SepSet_i(X, Y)$. 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 $\mathcal{K}$. 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 $\mathcal{K}$. Procedure possibleMConnectingPaths is again implemented as an extension to the algorithm in [15], with alterations similar to those mentioned in the previous paragraph for possible inducing paths.

```
Function MSeparationConstraints($X, Y, \mathbf{Z}, \mathcal{K}$)
$\Phi_c \leftarrow \varnothing$;
paths$\leftarrow$possibleMConnectingPaths($X, Y, \mathbf{Z}, \mathcal{K}$);
for each path $\in$ paths do
    $\Phi_c \leftarrow \Phi_c \wedge \neg mconnecting(path, \mathbf{Z})$;
end
return CNF($\Phi_c$)
```

Once the problem is instantiated, the constraints described in Proposition 4.3.1 are investigated using paths that admit a configuration that violates the constraints. These constraints include (only) the main variables of initial graphs, hence, any truth-setting assignment can be directly interpreted as a fully oriented graphical model. Parts of the constraints that are either TRUE or FALSE are omitted from the final sentence, since they take no part in the decision variable assignment. For example, In $\mathcal{K}$ depicted in Figure 4.2d, from $\mathcal{P}_1$, it occurs that B and F are independent condition to C. Therefore, path

$\langle B, C, F\rangle$ must be blocked condition to D. The corresponding constraint is $\neg edge(B,C) \vee \neg edge(C,F) \vee collider(B,C,F)$. However, $\{B,C,F\}$ is marked as a definite non-collider in $\mathcal{P}_1$ therefore only $\neg edge(B,C) \vee \neg edge(C,F)$ need to be included as a constraint.

Up to this point, the algorithm has included constraints that ensure that any graphical model resulting from FCM shares the same marginal independence models as $\mathcal{M}_T[_{\overline{\mathbf{O}_i}}$. In the attempt to choose a fitting model, the algorithm may orient both endpoints of an edge as tails, or orient a triple marked as a definite non-collider as a collider. Function `AdditionalConstraints` describes the relative constraints. The resulting formula is already in CNF, and no transformation is needed.

```
Function AdditionalConstraints($\mathcal{K}$)
$\Phi_c \leftarrow \varnothing$;
for each edge $(X,Y) \in \mathcal{K}$ do
    $\Phi_c \leftarrow \Phi_c \wedge (arrowhead(X,Y) \vee arrowhead(Y,X))$
end
for each triple $(X,Y,W) \in \mathcal{K}$ marked as definite non collider do
    $\Phi_c \leftarrow \Phi_c \wedge (\neg arrowhead(X,Y) \vee \neg arrowhead(W,Y))$
end
return $\Phi_c$
```

Hence, the graphical model corresponding to a variable assignment that satisfies the SAT instance built so far is a mixed graph that shares the same marginal independence models (according to the criterion of m-separation) as $\mathcal{M}_T[\overline{\mathbf{O}}_i$ for every i.

In several cases, the user may be interested in a MAG that is consistent with the data and has some specific properties. For example, the user may be interested in the possible causal relations between two specific variables, in order to plan an experiment. To satisfy such purposes, Algorithm 2 can be augmented with a query. The implemented version of the algorithm accepts queries referring to the existence of a causal path(of any length), but any kind of query that can be expressed in graphical terms is easily implemented. The query comprises the initial formula, which is then conjuncted with the remaining constraints. Thus, the model chosen is forced to include the desired characteristics. If no such model exists, the algorithm returns null.

Finally, we have to ensure that the model is an ancestral. One approach would be to include constraints preventing directed and almost directed cycles in the SAT formula. However, it proves computationally more efficient to check the model for directed and almost directed cycles after it has been returned. In cases cycles are detected, the SAT instance is conjuncted with additional constraints that prevent the edge and the formula is submitted to the SAT-solver again. This procedure is repeated until a valid model is returned (or NULL , if there is no valid model with the specified properties).

The SAT solver assumed returns FALSE, if the input formula is unsatisfiable. If the formula is satisfiable, it returns the truth-setting assignment of the variables. The CNF instance used as input to the SAT solver is equivalent to initial SAT that contains only the main variables of the instance. So, the transformation from a variable assignment to a graphical model makes use of these variables and only. The transformation is applied by function `makeChanges`.

Function `MakeChanges`$(\mathcal{K}, L)$

```
for each edge $(X,Y) \in \mathcal{K}$ do
    if $edge(X,Y) \in L^{-}$ then
        Remove edge from $\mathcal{K}$
    end
end
for circle endpoint $Xo - *Y$ in $\mathcal{K}$ do
    if $arrowhead(X,Y) \in L^{+}$ then
        Orient circle as an arrow
    end
    else
        Orient circle as a tail
    end
end
return $\mathcal{K}$
```

What remains to be done is check for maximality. We already ensured that the returned model is ancestral. Also, for edges we have encountered in at least one $\mathcal{P}_i$, we have included the constraints necessary to exclude the edge only if there is no primitive inducing path between its endpoints. For edges that are missing from the initial graph, maximality is ensured by the constraints added during execution of `MSeparationConstraints`. Thus, the remaining edges required to be checked are edges connecting variables that are never encountered together. The edges are checked, and , if necessary, the required bi-directed edges are added to make the graph maximal. Thus, the resulting graph is maximal, ancestral, and shares the same marginal m-separation structures as those observed in the data.

4.3.3 Soundness and Completeness

The following theorems prove that algorithm FCM is both sound and complete. For the proof, we only consider FCM with an empty causal query, which does not affect the validity of the proof. The only difference when including a query is that there exists the possibility that the formula is unsatisfiable. On the contrary, when the model is not forced to abide some specific (possibly mistaken) properties, there always exists a model that fits the observed data (the true causal MAG $\mathcal{M}_T$).

Theorem 4.3.1 *(soundness) Let $\mathcal{M}$ be the mixed graph resulting from algorithm 2 according to the variable assignment L, and let $\mathcal{M}_T$ be the actual data - generating MAG, corresponding to the actual independence model $\mathcal{J}$. Then the following holds:*
$\forall i, \mathcal{M}_T[_{\overline{\mathbf{O}_i}}$ and $\mathcal{M}[_{\overline{\mathbf{O}_i}}$ are Markov equivalent, i.e. they share the same edges, the same unshielded colliders and the same discriminating colliders.

Proof Obviously, $\mathcal{P}_i \in [\mathcal{M}_T[_{\overline{\mathbf{O}_i}}]$ for every i. If A, B are adjacent in $\mathcal{M}_T$, A, B are adjacent in $\mathcal{M}$ or there exists in $\mathcal{M}$ an inducing path from A to B w.r.t. $\overline{\mathbf{O}_i}$, therefore A and B are adjacent in $\mathcal{M}[_{\overline{\mathbf{O}_i}}$. If A, B are not adjacent in $\mathcal{M}_T[_{\overline{\mathbf{O}_i}}$, $\langle A, B|\mathbf{Z}\rangle \in \mathcal{J}[_{\overline{\mathbf{O}_i}}$, for some $\mathbf{Z} \subset \overline{\mathbf{O}_i}$ and Function `SeparatingConstraints` ensures that A and B are m-separated condition to $\mathbf{Z}$ in $\mathcal{M}$, therefore A, B are not adjacent in $\mathcal{M}[_{\overline{\mathbf{O}_i}}$. So, for every i, $\mathcal{M}_T[\overline{\mathbf{O}_i}$ and $\mathcal{M}[\overline{\mathbf{O}_i}$ share the same edges.

If (A, B, C) form an unshielded collider in $\mathcal{M}_T[_{\overline{\mathbf{O}_i}}$, A, B, C form an unshielded triple in $\mathcal{M}[_{\overline{\mathbf{O}_i}}$ and $\langle A, B|\mathbf{Z}\rangle \notin \mathcal{J}[_{\overline{\mathbf{O}_i}}$ for any $\mathbf{Z} \subset \mathbf{O_i}$ that contains B, therefore (A, B, C) forms a collider in $\mathcal{M}[_{\overline{\mathbf{O}_i}}$. Similarly, if (A, B, C) forms an unshielded non-collider in $\mathcal{M}_T[_{\overline{\mathbf{O}_i}}$, $\langle A, B|\mathbf{Z}\rangle \in \mathcal{J}[_{\overline{\mathbf{O}_i}}$ for some $\mathbf{Z} \subset \mathbf{O_i}$ that contains B, therefore (A, B, C) forms a non-collider in $\mathcal{M}[_{\overline{\mathbf{O}_i}}$.

If $p = \langle X, \ldots, W, V, Y\rangle$ is a discriminating path in both $\mathcal{M}_T[_{\overline{\mathbf{O}_i}}$ and $\mathcal{M}[_{\overline{\mathbf{O}_i}}$, if V is a collider in $\mathcal{M}_T[_{\overline{\mathbf{O}_i}}$ $\langle X, Y|\mathbf{Z}\rangle$ is not a member of $\mathcal{J}[_{\overline{\mathbf{O}_i}}$ for any $\mathbf{Z} \subset \mathbf{O}_i$ that includes V, and since p must be blocked condition to $\mathbf{Z}$, V must be a collider on p. Similarly if V is not a collider in $\mathcal{M}_T[_{\overline{\mathbf{O}_i}}$ $\langle X, Y|\mathbf{Z}\rangle$ is a member of $\mathcal{J}[_{\overline{\mathbf{O}_i}}$ for some $\mathbf{Z} \subset \mathbf{O}_i$ that includes V, and since p must be blocked condition to $\mathbf{Z}$, V must be a non-collider on p.

Theorem 4.3.2 *(completeness) Let $\mathcal{M}_T$ be a MAG consistent with all $\mathcal{P}_i$, i.e. $\mathcal{P}_i \in [\mathcal{M}_T[_{\overline{\mathbf{O}_i}}]$ for every i. Then $\mathcal{M}_T$ corresponds to a truth-setting assignment for the SAT instance generated by Function `GenerateConstraints`.*

Proof If X, Y are adjacent in $\mathcal{P}_i$, either X, Y adjacent in $\mathcal{M}_T$ therefore $edge(X, Y)$ is TRUE or there exists an inducing path from X to Y with respect to $\overline{\mathbf{O}_i}$ there for for some path $inducing(path, \overline{\mathbf{O}_i})$ is TRUE. Moreover, $\mathcal{M}_T$ is maximal, therefore if X, Y are adjacent in $\mathcal{M}_T$, there exists no inducing path from X to Y w.r.t $\varnothing$. Thus, all constraints added by AdjacencyConstraints are satisfied.

If X, Y are not adjacent in $\mathcal{P}_i$, then there is no m-connecting path from X to Y condition to $SepSet_i(X, Y)$, thus, all constraints added by Function `SeparatingConstraints` are satisfied.

If a triple forms a definite non-collider in some $\mathcal{P}_i$, the triple cannot be a collider in $\mathcal{M}_T$. Finally, $\mathcal{M}_T$ can not contain any edges with two tails. Thus, constraints added by Function `AdditionalConstraints` are satisfied.

Thus, any MAG consistent with all $\mathcal{P}_i$ satisfies the SAT instance generated by Function `GenerateConstraints`.

4.4 The Pairwise Causal Graph

We have presented an algorithm that returns a specific MAG consistent with a set of marginal distributions. 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 4.4.1 *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:*

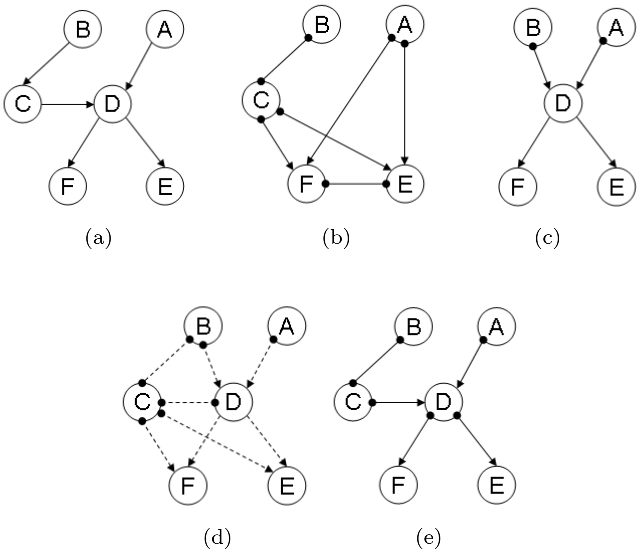
1. *X -- Y in $\mathcal{U}$ if X is adjacent to Y in at least one $\mathcal{M}$ consistent with all $\mathcal{P}_i$.*

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.*

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.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. However, they clearly illustrate the possible pairwise causal relationships, and they often approximate the PAG representing the Markov Equivalence class of the data-generating MAG very well.

The Causal SAT algorithm (cSAT) repeatedly invokes FCM with a causal query for every uncertainty present in $\mathcal{K}$ resulting from line 3 of Algorithm 2. Each rejected query is imposed on $\mathcal{K}$ which is returned as the output Pairwise Causal Graph. A detailed example of how algorithm cSAT progresses is shown in Figure 4.2



(a) (b) (c)

(d) (e)

Figure 4.2: 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.

Algorithm 9: cSAT

```
Input: ${\mathcal{P}_i}_{i=1}^N$
Output: $\mathcal{U}$, the Pairwise Causal Graph over $\mathbf{O} = \bigcup_{i=1}^N \mathbf{O}_i$
$\mathcal{U} \leftarrow$ InitializeGraph ($\mathcal{P}$);
for every edge $X, Y$ in $\mathcal{U}$ do
    if FCM $(\{\mathcal{P}_i\}_{i=1}^N, edge(X,Y)) == \varnothing$ then
        Remove edge from $\mathcal{U}$
    end
    else if FCM $(\{\mathcal{P}_i\}_{i=1}^N, \neg edge(X,Y) == \varnothing$ then
        Mark edge as solid
    end
end
for every unoriented endpoint $X * \cdots \circ Y$ in $\mathcal{U}$ do
    if FCM $(\{\mathcal{P}_i\}_{i=1}^N, edge(X,Y) \wedge arrowhead(X,Y)) == \varnothing$ then
        Orient $X$ out of $Y$
    end
    else if FCM $(\{\mathcal{P}_i\}_{i=1}^N, edge(X,Y) \wedge \neg arrowhead(X,Y)) == \varnothing$ then
        Orient $X$ into $Y$
    end
end
return $\mathcal{U}$
```

4.5 Algorithm cSAT+

The initial graph created by Function 3 of Algorithm 9 contains several unoriented edges, connecting variables that have not been measured together. These edges are combined in different ways forming numerous possible inducing an m-connecting paths, which are transformed to boolean constraints exhaustively in the latter steps of the algorithm. Some of these edges, however, can be removed or oriented based on a simpler reasoning. the following two propositions are used as preprocessing steps:

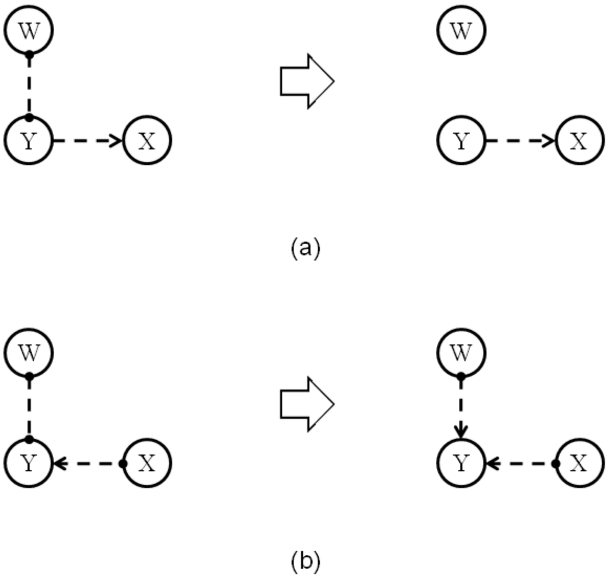


Figure 4.3: Preprocessing steps 1 and 2. W and X have been found conditionally independent in one dataset, while Y and X adjacent in another.

Proposition 4.5.1 (Preprocessing Step 1) *If $X \leftarrow Y$ in $\mathcal{P}_i$, for every m-separation $MSep(X, W|\mathbf{Z})$ in $\mathcal{P}_j$, $Y \notin \mathbf{O}_j$ with $\mathbf{Z} \cap \overline{\mathbf{O}_i} = \varnothing$, remove Y∘--∘W from $\mathcal{U}$.*

Proof Let $\mathcal{M}_T$ denote the (any) data-generating MAG over variables $\bigcup_i \mathbf{O}_i$. If $X \leftarrow Y$ in $\mathcal{P}_i$, X and Y cannot be m-separated by any subset of $\mathbf{O}_i$, therefore there exists in $\mathcal{M}_T$ an m-connecting path from Y to X condition to $\mathbf{Z}$(since $\mathbf{Z} \cap \overline{\mathbf{O}_i} = \varnothing$), that does not go through W. If W and Y are adjacent, the concatenation of edge (W, Y) with this path would be m-connecting W and X condition to $\mathbf{Z}$. Therefore, W and Y cannot be adjacent.

Proposition 4.5.2 (Preprocessing Step 2) *If $X \rightarrow Y$ in $\mathcal{P}_i$, for every m-separation $MSep(X, W|\mathbf{Z})$ in $\mathcal{P}_j$ with $\mathbf{Z} \cap \overline{\mathbf{O}_i} = \varnothing$, orient Y∘--∘W as Y ←- ∘W in $\mathcal{U}$*

Proof Similarly, if X is into Y in $\mathcal{P}_i$, there exists an m-connecting path from X to Y in $\mathcal{M}_T$ that does not go through W. Again, if W is out of Y, the concatenation of edge (W, Y) with this path would be m-connecting W and X condition to $\mathbf{Z}$.

A schematic representation of the preprocessing steps is shown in Figure 4.5 Algorithm cSAT is augmented with the two preprocessing steps to form Algorithm cSAT+.

Algorithm 10: cSAT+

```
Input: $\{\mathcal{P}_i\}_{i=1}^N$
Output: $\mathcal{U}$, the Pairwise Causal Graph over $\mathbf{O} = \bigcup_{i=1}^N \mathbf{O}_i$
$\mathcal{U} \leftarrow$ InitializeGraph ($\mathcal{P}$);
repeat
    Apply preprocessing steps 1 and 2;
until no step is applicable ;
for every edge $X, Y$ in $\mathcal{U}$ do
    if FCM ($\{\mathcal{P}_i\}_{i=1}^N, edge(X, Y)) == \varnothing$ then
        Remove edge from $\mathcal{U}$
    end
    else if FCM ($\{\mathcal{P}_i\}_{i=1}^N, \neg edge(X, Y) == \varnothing$ then
        Mark edge as solid
    end
end
for every unoriented endpoint $X * \cdots \circ Y$ in $\mathcal{U}$ do
    if FCM ($\{\mathcal{P}_i\}_{i=1}^N, edge(X, Y) \wedge arrowhead(X, Y)) == \varnothing$ then
        Orient $X$ out of $Y$
    end
    else if FCM ($\{\mathcal{P}_i\}_{i=1}^N, edge(X, Y) \wedge \neg arrowhead(X, Y)) == \varnothing$ then
        Orient $X$ into $Y$
    end
end
return $\mathcal{U}$
```

Chapter 5

Results

5.1 Evaluation of Inference Capabilities

. We empirically quantify the inference capability of cSAT+ on 7 networks in the literature [14, 13, 2]. 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[9] 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. Figure 5.1 clearly shows the inferential advantages gained by integrative analysis: most inference rates are significantly

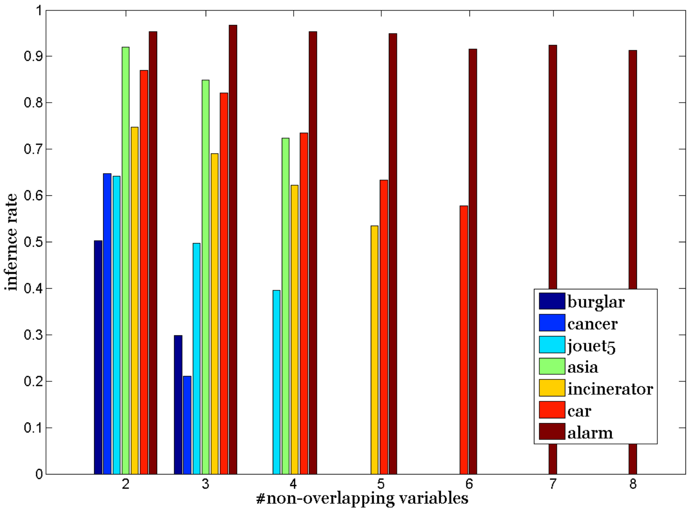


Figure 5.1: Running time for cSAT+ on 7 Networks.

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.

5.2 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 5.2 presents the ratio of the median number of 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.

5.3 Comparison with ION

We compare the algorithm with ION. ION [27] is a similar but more general algorithm, that produces all PAGs corresponding to a series of overlapping datasets. Tables 5.1-5.4 present 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

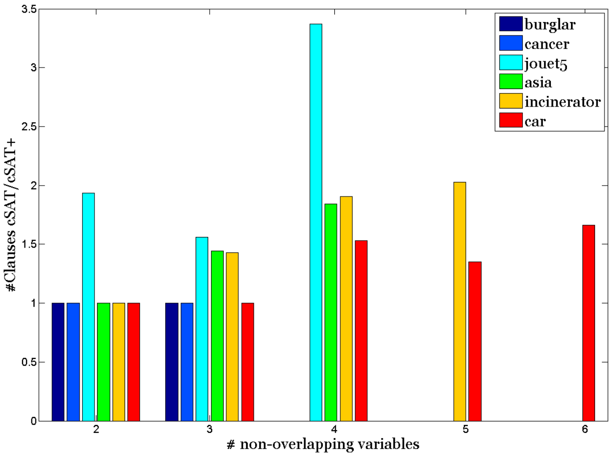


Figure 5.2: Median Clauses for cSAT+/Median Clauses for cSAT on 7 Networks.

enumerates all fitting PAGs taking up to 2 orders of magnitude more time than cSAT+.

5.4 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 5.3. 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.

5.5 Conclusions and Future Work

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

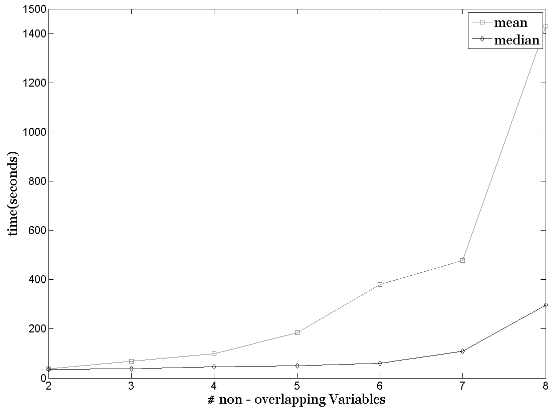


Figure 5.3: Running time for cSAT+ on ALARM Network.

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.

Scaling up to a medium-sized network such as ALARM, indicates that when the number of non-overlapping variables is small relative to the total size of the variables measured, the result is very close to the actual network, i.e. one can infer the correct independence model by observing overlapping marginal distributions.

These preliminary results are encouraging to further develop the methods to scale to larger and more realistic sizes. Similar algorithms can also be used to combine local structure knowledge for very large networks. Special methods for visualization of such models that focus on their probabilistic characteristics are still underdeveloped. Finally, the augmentation the algorithm with heuristic steps to deal with situations of imperfect knowledge of independencies (statistical errors) is of major interest.

Network	CANCER		BURGLAR		JOUET5		
# non-ov.Vars	2	3	2	3	2	3	4
Ion	0.0259	0.0340	0.0746	1.5179	0.3706	2.3055	-
cSAT	0.2141	0.3553	0.2571	0.8274	0.3741	0.6205	1.5799
cSAT+	0.1942	0.3216	0.2571	0.8274	0.3296	0.5406	0.9843

Table 5.1: Median running times for Ion, cSAT and cSAT+ on networks CANCER, BURGLAR and JOUET5.

Network	ASIA		
# non-ov.Vars	2	3	4
Ion	10.4729	64.0614	-
cSAT	0.4556	0.8852	1.8175
cSAT+	0.3754	0.6326	1.3247

Table 5.2: Median running times for Ion, cSAT and cSAT+ on network ASIA.

Network	INCINERATOR			
# non-ov.Vars	2	3	4	5
Ion	24.3025	-	-	-
cSAT	0.7142	1.3563	3.3828	7.9709
cSAT+	0.6557	1.1555	2.4885	4.6206

Table 5.3: Median running times for Ion, cSAT and cSAT+ on network INCINERATOR.

Network	CAR				
# non-ov.Vars	2	3	4	5	6
Ion	8.6150	330.2888	-	-	-
cSAT	0.4068	0.6713	1.1541	4.1449	2.6990
cSAT+	0.3860	0.6157	0.9133	2.1991	2.3288

Table 5.4: Median running times for Ion, cSAT and cSAT+ on network CAR.

Bibliography

[1] Ayesha R. Ali and Thomas Richardson. Markov equivalence classes for maximal ancestral graphs. In *UAI*, pages 1–9, 2002.

[2] I. Beinlich, G. Suermondt, R. Chavez, and G. Cooper. The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. In *2nd European Conference in Artificial Intelligence in Medicine*, pages 247–256, 1989.

[3] Richard A. Caruana. Multitask learning: A knowledge-based source of inductive bias. In *Proceedings of the Tenth International Conference on Machine Learning*, pages 41–48. Morgan Kaufmann, 1993.

[4] David Maxwell Chickering. Learning bayesian networks is np-complete. In *Learning from Data: Artificial Intelligence and Statistics V*, pages 121–130. Springer-Verlag, 1996.

[5] David Maxwell Chickering and Craig Boutilier. Optimal structure identification with greedy search, 2002.

[6] Gregory F. Cooper and Tom Dietterich. A Bayesian method for the induction of probabilistic networks from data. In *Machine Learning*, pages 309–347, 1992.

[7] David Danks. Learning the causal structure of overlapping variable sets. In *Discovery Science*, pages 178–191, 2002.

[8] David Danks. Learning integrated structure from distributed databases with overlapping variables. Technical Report CMU-PHIL-149, Department of Philosophy, Carnegie Mellon University, 2003.

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

[10] R.A Fisher. *The Design Of Experiments,*. London,Oliver and Boyd,, 1960.

[11] Dan Geiger and Judea Pearl. Logical and algorithmic properties of independence and their application to bayesian networks. *Ann. Math. Artif. Intell.*, 2:165–178, 1990.

[12] C W J Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37(3):424–38, 1969.

[13] K. Korb and A. E. Nicholson. *Bayesian Artificial Intelligence*. Chapman & Hall /CRC, 2003.

[14] S. L. Lauritzen and D.J. Spiegelhalter. Local computation with probabilities on graphical structures and their application to expert systems (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 50(2):157–224, 1988.

[15] R. E. Neapolitan. *Learning Bayesian Networks*. Prentice Hall, 2003.

[16] Keith O'Rourke. An historical perspective on meta-analysis: dealing quantitatively with varying study results. *J R Soc Med*, 100(12):579–582, 2007.

[17] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. Technical report, Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong, China, November 2008.

[18] J. Pearl. *Causality, Models, Reasoning, and Inference*. Cambridge University Press, Cambridge, U.K., 2000.

[19] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.

[20] Judea Pearl and T.S. Verma. A theory of inferred causation, 1991.

[21] H. Reichenbach. *The Direction of Time*. University of California Press, Berkeley, 1956.

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

[23] Moninder Singh and Marco Valtorta. Construction of Bayesian network structures from data: A brief survey and an efficient algorithm. *Int. J. Approx. Reasoning*, 12(2):111–131, 1995.

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

[25] P. Spirtes, C. Meek, and T. Richardson. An algorithm for causal inference in the presence of latent variables and selection bias. In *Computation, Causation, and Discovery*, pages 211–252. AAAI Press, Menlo Park, CA, 1999.

[26] P. Spirtes and Th. Richardson. A polynomial time algorithm for determining DAG equivalence in the presence of latent variables and selection bias, 1997.

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

[28] Robert E. Tillman. Structure learning with independent non-identically distributed data. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1041–1048. ACM, 2009.

[29] I. Tsamardinos, L. E. Brown, and C. F. Aliferis. The max-min hill-climbing bayesian network structure learning algorithm. *Machine Learning*, 65(1):31–78, 2006.

[30] Ioannis Tsamardinos and Asimakis P. Mariglis. Multi-source causal analysis: Learning bayesian networks from multiple datasets. In *AIAI*, pages 479–490, 2009.

[31] Thomas Verma and Judea Pearl. Causal networks: semantics and expressiveness. In *UAI*, pages 69–78, 1988.

[32] Thomas Verma and Judea Pearl. Equivalence and synthesis of causal models. In *UAI*, pages 255–270, 1990.

[33] J. Zhang. Causal reasoning with ancestral graphs. *J. Mach. Learn. Res.*, 9:1437–1474, 2008.

[34] J. Zhang. 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, 2008.