Aix*Marseille université
Socialement engagée

NNT : 2025AIXM0017

THÈSE DE DOCTORAT

Soutenue à Aix-Marseille Université
le 09 janvier 2025 par

Amir Hosein VALIZADEH

Causal Bayesian Network Learning with Latent Confounders Identification and its Applications to a Medical Problem

Discipline
Informatique

École doctorale
ED 184
Mathématiques et Informatique

Laboratoire/Partenaires de recherche
LIS - Laboratoire d'Informatique et des Systèmes
(CNRS UMR 7020)

Composition du jury

Philippe LERAY Professeur - Nantes Université	Rapporteur
Nahla BEN AMOR Professeure - Institut Supérieur de Gestion de Tunis	Rapporteure
Simon DE GIVRY CR HDR - INRAe Toulouse	Examinateur
Jonathan MESSIKA PU-PH - Université de Paris Cité	Président du jury
Christophe GONZALES Professeur - Aix-Marseille Université	Directeur de thèse
Stéphane DELLIAUX MCU-PH, HDR - Aix-Marseille Université	Co-encadrant de thèse
Stéphane BURTEY PU - PH - Aix-Marseille Université	Co-directeur de thèse

cnrs

LIS
LABORATOIRE D'INFORMATIQUE & DES SYSTÈMES
UMR 7020

Affidavit

I, undersigned, Amir Hosein VALIZADEH, hereby declare that the work presented in this manuscript is my own work, carried out under the scientific supervision of Christophe GONZALES, and co-supervision of Stéphane DELLIAUX and Stéphane BURTEY in accordance with the principles of honesty, integrity and responsibility inherent to the research mission. The research work and the writing of this manuscript have been carried out in compliance with both the French National Charter for Research Integrity and the Aix-Marseille University Charter on the fight against plagiarism.

This work has not been submitted previously either in this country or another country in the same or a similar version to any other examination body.

Marseille 25/09/2024



This work is licensed under Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International Public License

Résumé

Cette thèse explore les avancées dans l'apprentissage causal au sein des Réseaux Bayésiens (BNs), avec un accent particulier sur les défis posés par les variables latentes et les complexités de l'analyse des données médicales. Elle introduit une approche basée sur le score pour identifier les variables latentes, en utilisant des motifs triangulaires dans les Graphes Acycliques Dirigés (DAGs) pour améliorer la précision et l'interprétabilité des inférences causales. Cette méthodologie a été validée à travers des simulations étendues et montre un potentiel prometteur pour améliorer la précision, le rappel et les scores F1, en particulier dans des scénarios impliquant de grands ensembles de données. Importamment, les techniques développées pour l'identification des variables latentes sont intégrées dans l'analyse des données médicales, où elles sont appliquées pour renforcer la construction des modèles de Réseaux Bayésiens pour les issues des chirurgies de transplantation pulmonaire. Cette approche intégrée aborde les problèmes courants de petites tailles d'ensembles de données et de données manquantes en employant des algorithmes de bootstrap et d'Espérance-Maximisation (EM), spécifiquement adaptés aux contextes des données médicales. Le modèle de Réseau Bayésien résultant, qui identifie des relations causales significatives, est également soutenu par des niveaux de confiance et des validations d'experts, soulignant la robustesse et l'applicabilité de nos méthodes. Ce travail vise à faire le pont entre des méthodes probabilistes et statistiques avancées et des applications pratiques en science médicale, fournissant des outils et des perspectives qui pourraient être utiles dans le domaine de l'apprentissage causal. Il suggère des applications potentielles dans le secteur de la santé et au-delà, notant les rôles possibles que ces outils pourraient jouer dans le soutien au diagnostic et à l'élaboration des politiques. En fin de compte, cette thèse cherche à contribuer à la discussion continue sur les systèmes complexes et leur dynamique causale, dans l'espoir d'offrir des insights qui pourraient s'avérer précieux dans divers domaines.

Mots-clés: Réseaux Bayésiens Causaux, Apprentissage de Structure Basé sur le Score, Variables Latentes, Analyse de Données Médicales, Algorithmes d'Espérance-Maximisation, Techniques de Bootstrap

Abstract

This thesis investigates advancements in causal learning within Bayesian Networks (BNs), with a particular focus on addressing the challenges posed by latent variables and the complexities of medical data analysis. It introduces a score-based approach for identifying latent variables, utilizing triangle patterns within Directed Acyclic Graphs (DAGs) to enhance the accuracy and interpretability of causal inferences. This methodology has been validated through extensive simulations and shows promise in improving precision, recall, and F1 scores, particularly in scenarios involving larger datasets. Importantly, the techniques developed for latent variable identification are integrated into the analysis of medical data, where they are applied to enhance the construction of Bayesian Network models for lung transplant surgery outcomes. This integrated approach addresses common issues such as small dataset sizes and missing data by employing bootstrapping and Expectation Maximization (EM) algorithms tailored specifically for medical contexts. The resultant Bayesian Network model, which identifies significant causal relationships, is further supported by confidence levels and expert validations, underscoring the robustness and applicability of our methods. This work aims to bridge advanced probabilistic and statistical methods with practical applications in medical science, providing tools and perspectives that could be useful in the field of causal learning. It suggests potential applications within healthcare and beyond, noting the possible roles these tools might play in supporting diagnostics and policy-making. Ultimately, this thesis seeks to contribute to the ongoing discussion about complex systems and their causal dynamics, hoping to offer insights that may prove valuable in various fields.

Keywords: Causal Bayesian Networks, Score-Based Structure Learning, Latent Variables, Medical Data Analysis, Expectation Maximization Algorithms, Bootstrapping Techniques

Aknowledgements

I would like to express my deepest gratitude to all the individuals who contributed to the completion of this thesis.

First and foremost, I wish to warmly thank my supervisor, Professor Christophe Gonzales, for his exceptional guidance, constant availability, and invaluable advice throughout these years. His expertise in computer science and unwavering support were crucial to the success of this work.

I would also like to express my appreciation to co-supervisors, Professors Stéphane Delliaux and Stéphane Burtey for their medical expertise, whose frequent meetings and dedicated support greatly enriched this research. Thanks them for their advice and assistance.

I sincerely thank the members of COALA team at LIS lab, particularly Professors Cyril Terrioux, Djamal Habet, and Philippe Jégou for their valuable advice and technical support. I am also grateful to my fellow PhD students, especially Sami Cherif, Haseena Sreedharan, Roza Amokrane, and Yousra El Ghazi, for their constructive discussions and camaraderie.

I sincerely thank the members of my jury, including Nahla Ben Amor, Philippe Leray, Simon De Givry, and Jonathan Messika, for dedicating their valuable time and effort to thoroughly review my thesis. Their insightful comments, detailed feedback, and constructive suggestions, both in their written reports and during my defense session, significantly enhanced the quality and clarity of my dissertation. I am truly grateful for their guidance and expertise, which have helped me to refine my work substantially.

Finally, I wish to thank my family from the bottom of my heart. Thank you to my parents, Farideh and Alireza, for their enduring support and constant encouragement. To my brothers, Meysam and Maziar, thank you for your understanding and moral support. Most importantly, I want to express my deepest gratitude to my dear wife, Narges, whose boundless love, extraordinary patience, and continuous encouragement have been my greatest source of strength throughout this journey. This accomplishment would not have been possible without you.

To each of you, I am profoundly grateful, and sincerely appreciate everything each of you has done for me.

Contents

Affidavit 2

Résumé 3

Abstract 4

Aknowledgements 5

Contents 6

List of Figures 9

List of Tables 11

Introduction 12

I. State of the Art 16

1. Bayesian Networks . . . 16
   - 1.1. Graph Terminology . . . 17
   - 1.2. Basic Definitions . . . 18
   - 1.3. Representation . . . 21
2. Causal Networks . . . 26
   - 2.1. Essential Concepts . . . 26
     - 2.1.1. Correlation vs. Causation . . . 26
   - 2.2. Causal Models . . . 28
     - 2.2.1. Latent Variables . . . 29
     - 2.2.2. Partially Directed Acyclic Graphs . . . 32
   - 2.3. Causal Inference . . . 34
     - 2.3.1. Observational vs. Interventional . . . 34
     - 2.3.2. do-calculus and Identifiability . . . 36
     - 2.3.3. Potential Outcomes . . . 40
   - 2.4. Causal Learning . . . 44
     - 2.4.1. Ancestral Graphs . . . 45
       - 2.4.1.1. Maximal Ancestral Graph (MAG) . . . 47
       - 2.4.1.2. Partial Ancestral Graph (PAG) . . . 48
     - 2.4.2. Causal Discovery . . . 49

3. Learning . . . 52
3.1. Parameter Learning . . . 52
3.1.1. Maximum Likelihood Estimation (MLE) . . . 53
3.1.2. Bayesian Estimation . . . 56
3.2. Structure Learning . . . 57
3.2.1. Constraint-based Approaches . . . 58
3.2.1.1. Causal Sufficient Algorithms . . . 59
3.2.1.2. Causal Insufficient Algorithms . . . 62
3.2.2. Score-based Approaches . . . 65
3.2.2.1. Scoring Functions . . . 66
3.2.2.2. Approximate Algorithms . . . 72
3.2.2.3. Exact Algorithms . . . 75
3.2.3. Hybrid Algorithms . . . 76

II. Contributions **79**

4. A New Full Score-based Causal Learning Algorithm . . . 81
4.1. Triangles in DAGs . . . 81
4.1.1. Genuine vs. Latent Triangles . . . 91
4.2. The Algorithm . . . 93
4.2.1. Triangles Discrimination . . . 93
4.2.2. Finding d-Separating Sets . . . 96
4.2.3. Modifying the Structure . . . 100
4.2.4. Complexity Analysis . . . 102
4.3. Experiments . . . 104
4.4. Conclusion . . . 116
5. A Causal Learning Approach for Medical Data . . . 117
5.1. Introduction . . . 117
5.2. Methodology . . . 118
5.2.1. Data Description . . . 118
5.2.1.1. Data Discretization and Variable Types . . . 119
5.2.2. Handling Missing Data . . . 120
5.2.3. Bootstrapping for Data Augmentation . . . 122
5.2.4. Constructing a Trustworthy Structure . . . 124
5.2.5. Weighted Causal Effects . . . 126
5.2.5.1. Shapley Values: Bridging the Quantitative Gap 127
5.2.5.2. Graphical Representation of Shapley Values . . 129
5.2.6. Latent Variables Identification . . . 132
5.3. Experiments . . . 132
5.3.1. Model Generation . . . 132
5.3.2. Validation of Bootstrapping Technique . . . 133
5.3.3. Bootstrapping and Structure Learning . . . 141
5.3.4. Results . . . 141
5.3.5. Discussion . . . 149
5.4. Conclusion . . . 149

III. Conclusion and Future Works **151**
6. Conclusion 151
7. Future Works 152

Liste de participation aux conférences **155**

Bibliography **156**

List of Figures

I.1. Graphs . . . 17
I.2. A Bayesian Network with CPDs . . . 25
I.3. Spurious Correlation . . . 27
I.4. Correlation is not Causation . . . 27
I.5. Different Models to Check if Smoking Cause Cancer . . . 28
I.6. DAG vs. Partially DAGs . . . 34
I.7. Identifiable Causal Effect $P(\mathbf{Y} \mid do(\mathbf{X}))$ . . . 38
I.8. Not Identifiable Causal Effect $P(\mathbf{Y} \mid do(\mathbf{X}))$ . . . 39
I.9. Causal Structures . . . 42
I.10. Ancestral Graphs . . . 45
I.11. Meek's 3rd rule . . . 60
I.12. Meek's 4rd rule . . . 61
I.13. V-structures Conflicts . . . 62

II.1. Triangles in a DAG . . . 82
II.2. Original BN ($\mathcal{G}^*$) . . . 83
II.3. Creation of a Triangle in a Score-based Learning Algorithm . . . 83
II.4. Indistinguishable Structures . . . 90
II.5. Genuine vs. Latent Triangles . . . 92
II.6. Possible Learned Latent Triangles . . . 94
II.7. Possible Learned Arcs Connected to the Latent Triangles . . . 94
II.8. d-separating set $\{E, F\}$ blocking the path between the vertices of a triangle 97
II.9. Graph Modification Steps of Algorithm 2 . . . 101
II.10. Converting a DAG to a CPDAG . . . 103
II.11. Comparative plots for "ok" and "¬ok" metric across different CBNs. . . . 109
II.12. Comparative plots for "Precision" and "Recall" metric across different CBNs. . . . 110
II.13. Comparative plots for "ok" and "miss" metric across different CBNs' learned CPDAGs. . . . 114
II.14. Comparative plots for "type" and "xs" metric across different CBNs' learned CPDAGs. . . . 115
II.15. Network with Weighted Arcs . . . 130
II.16. Results for 'ok' metric . . . 137
II.17. Results for 'type' metric . . . 138
II.18. Results for 'xs' metric . . . 139
II.19. Results for 'links' metric . . . 140

II.20.Primary learned CPDAG from the imputed original dataset 142
II.21.Weighted Causal Structure 144
II.22.Revised Weighted Causal Structure 146
II.23.Final Causal Structure with Latent Variable 148

List of Tables

I.1. Artificial Data for Disease Treatment 41
I.2. Constraint-based Structure Learning Algorithms 59
I.3. Score-based Structure Learning Algorithms 66
I.4. Scores Comparison 72
I.5. Hybrid Structure Learning Algorithms 77

II.1. Properties of the CBNs employed for the experiments 105
II.2. Confounders found for different dataset sizes and CBNs with 2 Boolean confounders 108
II.3. Comparisons of the learned CPDAGs with those of the CBNs (with confounders) that generated the datasets. 112
II.4. Metrics used in Table II.3 112
II.5. Temporal Order of Variables 119
II.6. Raw Shapley Values of the Network in Figure II.15 130
II.7. Normalized Shapley Values of the BN in Figure II.15 131
II.8. Comparisons of the primary and the bootstrapping learned structures, with the Original CBNs that generated the datasets. 134
II.9. Confidence levels for the the learned links ($\mathscr{L}_C$) 143

Introduction

In today's world, where the volume, variety, and velocity of data generation surpass anything previously known, understanding complex relationships within these datasets is increasingly central to scientific advancement. The data has the potential to provide valuable insights because it can help us make better decisions in different areas like healthcare, economics, and environmental management. It can also help us develop new technologies and scientific discoveries. Its value lies not just in its quantity but also in our ability to analyze, understand, and find meaningful relationships in it. This is where modern scientific research is focused: interpreting the data in detail to identify patterns and connections that are not immediately obvious. Finding hidden relationships and patterns in complex datasets is a challenging task. This thesis focuses on causal Bayesian networks, particularly their application in medical data analysis. By developing new learning approaches for finding causal relationships, we aim to provide a deeper understanding of disease dynamics, which is critical for effective diagnosis, treatment, and prevention strategies.

Probabilistic graphical models known as Bayesian networks have transformed the way researchers understand uncertainties and dependencies in complex systems. Being able to model complex stochastic processes makes Bayesian networks an extremely valuable tool in fields ranging from artificial intelligence to medical sciences. Moving from Bayesian networks to causal networks represents a paradigm shift. While Bayesian networks are being used for modeling probabilistic relationships, causal networks go a step further by inferring causal relationships between variables. In medical research, discerning causality—not merely correlation—provides crucial insights into disease mechanisms and the efficacy of treatments. Causal networks enable researchers to not only predict outcomes but also to understand the underlying mechanisms, answering 'why' something happens, not just 'what' happens. This thesis aims to contribute to this understanding by developing methodologies for discerning causal relationships in medical data, which may support more effective diagnosis, treatment, and preventative strategies.

Learning causal Bayesian networks from medical data poses significant challenges despite their potential for unraveling complex medical phenomena. One of the most problematic issues in applying causal Bayesian networks to medical data is identifying and handling latent variables. These variables, not directly observed or measured, are inferred from other observed variables and are crucial for understanding the complete picture of complex systems. They often represent underlying processes or factors significantly impacting the observable variables. The presence of latent

variables in a dataset can notably affect the accuracy of inferences made by causal networks. Ignoring these variables can lead to incorrect assumptions about causality and, consequently, erroneous conclusions—an unacceptable risk in medical research. Furthermore, the complexity and often noisy nature of medical data add layers of difficulty in constructing accurate and reliable causal models. This thesis endeavors to contribute to resolving these critical challenges by refining the process of causal network construction, aiming to enhance both the scientific robustness and practical applicability of these models in medical settings.

The fundamental objectives of this research are twofold yet integrally linked. Firstly, it seeks to contribute to the field of causal Bayesian network learning by proposing a novel method that identifies and incorporates latent variables into the network. This approach is designed to enhance the accuracy and interpretability of causal Bayesian networks, addressing a recognized limitation in current methodologies. The primary research question is: How can integrating latent variable identification into causal Bayesian network learning enhance these models' overall accuracy and reliability compared to existing approaches in the literature? Building on this, the research extends the application of this innovative latent variable approach to medical datasets, aiming to develop and validate a robust causal learning strategy tailored for medical scenarios. This strategy addresses common challenges, such as small dataset sizes and missing data, leveraging the insights gained from the latent variable identification to offer deeper insights into complex medical conditions. The main question here is: How might the integration of latent variable identification with bootstrapping and EM algorithms facilitate the construction of causal Bayesian networks from medical data? These questions underscore the research's progression from theoretical exploration to practical application, aiming to make a meaningful contribution to the field of causal Bayesian network learning and its implementation in medical science.

This thesis contributes to the domain of causal Bayesian network learning by introducing a score-based approach for identifying latent variables in discrete datasets and applying this methodology to the analysis of medical data. Designed to address common challenges such as missing data and limited sample sizes, this methodology balances theoretical robustness with practical applicability and has been evaluated through comprehensive simulations and analysis of real-world medical data. The findings of this research could provide valuable insights into understanding, diagnosing, and treating complex medical conditions, suggesting ways to bridge the gap between computational theory and its application in medical science. While these developments are promising, they represent initial steps toward deeper understanding and application, reflecting ongoing efforts to enhance the utility of causal Bayesian networks in medical research.

The methodology proposed in this thesis offers several innovations that contribute to the field of causal Bayesian network learning:

- **Exploring the Score-Based Approach for Latent Variables**: This research explores a novel application of score-based methods for detecting latent variables

in discrete datasets. This approach represents an important exploration beyond traditional methods and offers a more nuanced analysis in causal network learning.

- **Causal Learning of Small Data with Missing Values**: The study also seeks to enhance the methodology by integrating bootstrapping and EM algorithms tailored specifically to address the challenges of missing data and small sample sizes prevalent in medical data. While these techniques have been known and applied in various contexts, their innovative use here lies in the specific combination and application to meet the unique demands of modern medical research. This approach aims to significantly improve the robustness and reliability of causal inferences, which is crucial in settings where data limitations often hinder comprehensive analysis.
- **Aiming for Enhanced Accuracy and Practical Implications**: The methodologies developed in this thesis are intended to demonstrate improved performance and accuracy, potentially enabling a more reliable understanding of complex causal relationships, especially in medical data analysis. Such improvements could have important implications for developing better diagnostic tools and treatment strategies in the medical field.
- **Potential Practical Implications and Future Applications**: The improved performance of this method is expected to have practical implications. In medical data analysis, for instance, it could lead to a more accurate and reliable understanding of disease mechanisms. The approach may also have applications in other fields such as genetics and social sciences, where understanding discrete data is crucial.

The structure of this thesis is designed to guide the reader through a detailed exploration of the field. The first part offers a thorough state-of-the-art review encompassing Bayesian Networks, Causal Networks, and their learning processes. This section not only establishes the theoretical background but also contextualizes the current research within the broader scientific dialogue. The second part details our contributions, including the development and rigorous evaluation of a new method for latent variable identification, alongside the introduction of a tailored causal learning approach for medical data. This approach specifically addresses prevalent challenges such as missing data and small dataset sizes. The aim here is to illustrate the scientific rigor of our methodologies and their potential applicability in the medical field, particularly in understanding complex causal relationships within medical data.

Following our contributions, the thesis progresses to a section on future work and perspectives. This segment explores possible extensions and applications of the research findings, examining how the introduced methods can be further refined and the potential challenges that may arise in broader applications. Opportunities for new research directions that emerge from this work are also discussed.

The thesis ends with a conclusion that synthesizes the key findings, reflecting on the contributions to the field and discussing the broader implications of the work. This final section not only concludes the thesis but also underscores the potential significance and the intended lasting impact of the research on the field of causal Bayesian network learning and beyond. Each section is carefully crafted to build upon the previous, creating a cohesive and comprehensive narrative that leads the reader from a thorough understanding of existing knowledge to the frontiers of current research and future possibilities.

I. State of the Art

Sommaire

1. Bayesian Networks . . . 16
   - 1.1. Graph Terminology . . . 17
   - 1.2. Basic Definitions . . . 18
   - 1.3. Representation . . . 21
2. Causal Networks . . . 26
   - 2.1. Essential Concepts . . . 26
     - 2.1.1. Correlation vs. Causation . . . 26
   - 2.2. Causal Models . . . 28
     - 2.2.1. Latent Variables . . . 29
     - 2.2.2. Partially Directed Acyclic Graphs . . . 32
   - 2.3. Causal Inference . . . 34
     - 2.3.1. Observational vs. Interventional . . . 34
     - 2.3.2. do-calculus and Identifiability . . . 36
     - 2.3.3. Potential Outcomes . . . 40
   - 2.4. Causal Learning . . . 44
     - 2.4.1. Ancestral Graphs . . . 45
     - 2.4.2. Causal Discovery . . . 49
3. Learning . . . 52
   - 3.1. Parameter Learning . . . 52
     - 3.1.1. Maximum Likelihood Estimation (MLE) . . . 53
     - 3.1.2. Bayesian Estimation . . . 56
   - 3.2. Structure Learning . . . 57
     - 3.2.1. Constraint-based Approaches . . . 58
     - 3.2.2. Score-based Approaches . . . 65
     - 3.2.3. Hybrid Algorithms . . . 76

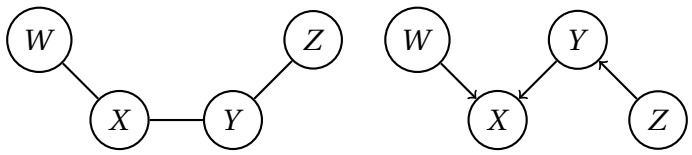
1. Bayesian Networks

Bayesian Network (BN) is one of the essential parts of probabilistic graphical models (PGMs) and provides an interesting and valuable framework for representing and reasoning about uncertain knowledge. A Bayesian Network is a directed acyclic graph (DAG) in which nodes represent random variables and may be interconnected by

edges. The absence of a direct edge between two nodes implies that they are conditionally independent. Since BNs are graphical models, reviewing some graph terminology before going through more details would be helpful (Section 1.1). Then we will present the basic definitions for the representation of Bayesian networks (Sections 1.2, and 1.3). Most of the definitions in the following section are from Koller and Friedman 2009. Hence, for further understanding of the concepts, it is recommended that the reader refers to it.

1.1. Graph Terminology

The term "graph" can refer to various visualizations. For instance, it can represent the plot of a function on a Cartesian coordinate system or a bar chart that displays categorical data using rectangular bars. However, this term pertains to a distinct set of mathematical objects in graph theory. These graphs comprise **nodes** or vertices interconnected by (undirected) **edges** (—). Figure I.1a illustrates an example of a graph with nodes W, X, Y, and Z and several edges connecting them. These graphs are called **undirected graphs** because their edges do not have any direction. Another type of graph consists of directed edges called **arcs** (⟶) instead of the undirected ones, and they are known as **directed graphs**. We use $\rightleftharpoons$ to represent the case where two nodes are connected, either by a directed (in any direction) or an undirected edge. Figure I.1b depicts a directed graph wherein an arc emanates from a **parent** node and goes into a **child** node. Two nodes are considered **neighbors** if connected by an edge. For example, in Figure I.1, X and Y are neighbors, but X and Z are not. The following definitions represent some principal concepts in the graphical models.



(a) Undirected Graph (b) Directed Graph

Figure I.1.: Graphs

Definition 1.1 (Graph) *A* Graph *is defined as a pair* $G = (\mathcal{X}, \mathcal{E})$ *where* $\mathcal{X}$ *is the set of* nodes, *and* $\mathcal{E}$ *is the set of* edges. *Whenever there is* $A \longrightarrow B$ *in* G, A *is a* parent *of* B, *and* B *is the* child *of* A. A *and* B *are the* neighbors *of each other when* $A \rightleftharpoons B \in \mathcal{E}$. *Sets are usually shown in* ***bold*** *letters. The parents, children, and neighbors of* A *are denoted as* $\boldsymbol{Pa}(A)$, $\boldsymbol{Ch}(A)$, *and* $\boldsymbol{Ne}(A)$, *respectively.*

Definition 1.2 (Skeleton) *A* skeleton *of a Graph* $G = (\mathcal{X}, \mathcal{E})$, *is its undirected version* $G' = (\mathcal{X}, \mathcal{E}')$ *where* $\mathcal{E}' = \{A—B | A \rightleftharpoons B \in \mathcal{E}\}$. *In other words, we would have a skeleton of a graph if the (undirected) edges replace the graph's arcs (directed edges).*

Definition 1.3 (Immorality) *A v-structure $A \longrightarrow C \longleftarrow B$ with no arc between A and B is considered as* immorality. *It is also called an unshielded collider. [Verma and Pearl 1991]*

Definition 1.4 (Clique) *A* clique, *or a complete subgraph, is a set of nodes $\mathcal{X}$ where every pair of nodes is connected by an edge.*

Definition 1.5 (Topological Order) *In a graph $G = (\mathcal{X}, \mathcal{E})$, an ordering like $\langle X_1, \cdots, X_k \rangle$ is denoted as a* topological order *regarding G if $X_i \longrightarrow X_j \in \mathcal{E}$ then $i < j$*

Definition 1.6 (Path) *In a graph $G = (\mathcal{X}, \mathcal{E})$, the sequence $X_1, \cdots, X_k$ of nodes is called a* path *if $X_i \rightleftharpoons X_{i+1}$ for every $i = 1, \cdots, k$*

Definition 1.7 (Directed Path) *A path $\langle X_1, \ldots, X_k \rangle$ is directed if for all $i \in \{1, \ldots, k-1\}$, the graph contains the arc $X_i \rightarrow X_{i+1}$.*

For instance, Figure I.1a shows a path of $< W, X, Y, Z >$ and $< Z, Y, X >$ is a directed path in Figure I.1b. If a directed path starts at node A and ends at node B, then A is an **ancestor** of B, and B is a **descendant** of A. Denote A's descendants and B's ancestors as **De**(A) and **An**(B). In Figure I.1b, Z is the ancestor of X and Y (**An**(X), **An**(Y)), while X and Y are both descendants of Z (**De**(Z)). If we add an arc from X to Z, then Z becomes an ancestor of itself. This structure is known as a directed **cycle**. A **directed acyclic graph** (**DAG**) has no directed cycles and no edges, while a **partially directed acyclic graph** (**PDAG**) has some (undirected) edges and some arcs (directed edges). A **completed partially directed acyclic graph** (**CPDAG**) can be created from a DAG by replacing all arcs with edges whenever they are not part of a v-structure, and only if reversing their direction will not create a new v-structure. CPDAGs are important for studying the Causal Networks, which we will explain in the other sections. Our thesis focuses mainly on DAGs and CPDAGs. In Section 2.2, we will discuss the differences among these graphs in more details.

1.2. Basic Definitions

After familiarizing some basic terms in the graphical models, we can go through fundamental Bayesian Networks' definitions. The main strength of BNs is their capacity to compactly represent joint probability distributions using local conditional probability distributions (CPDs). Each node in the BN contains a variable, while the graph structure encodes the probabilistic relationships between them.

The fundamental concept underlying BNs is conditional independence, called local independence, where each variable is independent of its non-descendants given its parents. This factorization property allows for efficient computation of complex probabilities, making BNs suitable for a wide range of applications. This section will present these key definitions in the study of Bayesian Networks.

Definition 1.8 (Sample Space & Event) *When conducting a random experiment, the* sample space *(denoted by Ω) is the complete set of all possible outcomes. The sample space size can vary, depending on the experiment's nature, and may be either finite or infinite. An* event *(represented by E) is a subset of the sample space comprising a collection of potential outcomes. An event may be empty, containing no potential outcomes; a singleton, containing only one potential outcome; or proper, containing more than one potential outcome. For instance, in a coin toss, the sample space is $\Omega = \{heads, tails\}$, and an event could be $E = \{\}$, $E = \{heads\}$, $E = \{tails\}$, or $E = \{heads, tails\}$.*

Definition 1.9 (Probability Distribution) Probability distribution *P over $(\Omega, \mathscr{F})$refers to a function that assigns real values to events in $\mathscr{F}$. For it to be considered a valid probability distribution, it must meet Kolmogorov axioms [Kolmogorov and Bharucha-Reid 2018]:*

1. $P(E) \in \mathbb{R},\ P(E) \geq 0 \qquad \forall E \in \mathscr{F}$
2. $P(\Omega) = 1$
3. *(σ-additivity) if $E_1, E_2, \cdots \in \mathscr{F}$ are disjoint sets, then:* $P(\bigcup_{i=1}^{\infty} E_i) = \sum_{i=1}^{\infty} P(E_i)$

As with any study in this particular discipline, one of our objectives is to determine the impact of happening of an event on the probability of another event's occurrence. To accomplish this task, it is necessary to present the definition of a principal concept in Probability Theory: **Conditional Probability**.

Definition 1.10 (Conditional Probability) *The probability of event B happening when event A has already occurred is called the* conditional probability *of B given A [Kolmogorov and Bharucha-Reid 2018]:*

$$P(B|A) = \frac{P(A \cap B)}{P(A)} \tag{I.1}$$

One immediate result from Definition 1.10 is $P(A \cap B) = P(A)P(B|A)$, which is usually called **chain rule** and can be generalized in the following definition.

Definition 1.11 (Chain Rule) *If $E_1, \cdots, E_n$ are events, then the following equation is the* chain rule*, which is an outcome of the conditional probability's definition:*

$$P(E_1 \cap \cdots \cap E_n) = P(E_1)P(E_2|E_1)\cdots P(E_n|E_1 \cap \cdots \cap E_{n-1}) \tag{I.2}$$

The above-mentioned definitions are based on events. To make the definitions in Probability Theory more practical, we need to represent them in mathematical form

and use numerical or categorical values. For this purpose, we use **random variables** (usually written in capital letters, i.e., X), which are mathematical quantization of random events' outcomes.

Definition 1.12 (Random Variables) *Given a probability space* $(\Omega, \mathscr{F}, P)$*, a* random variable X *is a measurable function* $X : \Omega \to \mathbb{R}$*. The function* X *is said to be measurable if for every Borel set* $B \subseteq \mathbb{R}$*, the pre-image* $X^{-1}(B) = \{\omega \in \Omega \mid X(\omega) \in B\}$ *is an element of* $\mathscr{F}$*, i.e.,* $X^{-1}(B) \in \mathscr{F}$*. This ensures that the probability of these pre-images can be well-defined by the probability measure* P*. Random variables can take either discrete or continuous values, depending on their distribution. [Billingsley 2017]*

In most cases, random variables are indicated using capital letters. If bold capital letters are used, they refer to a set of random variables. After getting familiar with the definition of a random variable, X, we can analyze the events' distribution that X can represent. To do this, we examine the distribution of these events, commonly known as the **marginal distribution** of the random variable X, denoted by $P(X)$. When we face scenarios that require us to analyze the values of multiple random variables, it becomes essential to consider the **joint distribution** over these variables. This enables us to examine and understand the events under consideration effectively.

Definition 1.13 (Joint & Marginal Probability Distribution) *The* joint probability distribution*, denoted as* $P(X_1, \cdots, X_n)$*, refers to the probability distribution of the entire set of random variables,* $\mathbf{X} = \{X_1, \cdots, X_n\}$*. On the other hand, the* marginal probability distribution *of a specific variable* X_i*, denoted as* $P(X_i), i \in \{1, \cdots, n\}$*, refers to the probability distribution of* X_i *alone. These two terms are related in that we can obtain the marginal probability distribution of* X *from the joint probability distribution by performing a process known as* marginalization*. This involves summing over all possible values of the other random variables in the set:* $\sum_{X \setminus \{X_i\}} P(X_1, \cdots, X_n) = P(X_i)$*. Obviously, if the variables are continuous, the sum needs to be replaced by a desired integral.*

Definition 1.14 (Statistical Independence) *Two random variables* X *and* Y *are* statistically independent *under a probability distribution* P *if the occurrence of a specific value of* X *does not influence the probability of* Y *taking any of its possible values, and vice versa, according to* P*. This is formally expressed as* $X \perp\!\!\!\perp_P Y$*, which means that for any values* x *of* X *and* y *of* Y*, the joint probability of* $X = x$ *and* $Y = y$ *is the product of their individual probabilities:*

$$P(X = x \cap Y = y) = P(X = x)P(Y = y)$$

Furthermore, if $P(Y = y) > 0$*, then* $P(X = x|Y = y) = P(X = x)$*, and if* $P(X = x) > 0$*, then* $P(Y = y|X = x) = P(Y = y)$*. Independence is symmetric, indicating that* $X \perp\!\!\!\perp_P Y$ *implies* $Y \perp\!\!\!\perp_P X$*.*

Definition 1.15 (Conditional Independence) *Two random variables X and Y are* conditionally independent *given a third random variable Z under a probability distribution P if the occurrence of a specific value of X does not influence the probability of Y taking any of its possible values, given the value of Z. This is denoted as $X \perp\!\!\!\perp_P Y \mid Z$ and means that for any values x of X, y of Y, and z of Z:*

$$P(X = x \text{ and } Y = y \mid Z = z) = P(X = x \mid Z = z)P(Y = y \mid Z = z)$$

This condition holds if and only if the joint probability of X and Y, given Z, equals the product of their conditional probabilities given Z. Equivalently, it can also be expressed as:

$$P(X = x|Y = y, Z = z) = P(X = x|Z = z)$$

and

$$P(Y = y|X = x, Z = z) = P(Y = y|Z = z)$$

for all x, y, and z where $P(Z = z) > 0$. Conditional independence assumes that any dependency between X and Y can be explained by their common relationship to Z.

1.3. Representation

The next step is to present a precise definition of the underlying principles of a Bayesian Network structure. This will help us to match the insight developed in previous sections to the formal presentations.

Definition 1.16 (Bayesian Network Structure) *A DAG called a* Bayesian Network structure $\mathcal{G}$ *, represents random variables $\mathbf{X} = \{X_1, \cdots, X_n\}$ in its nodes. The parents of X_i in $\mathcal{G}$ are denoted as $\boldsymbol{Pa}_{\mathcal{G}}(X_i)$, and the variables in the graph that are not descendants of X_i are denoted as $\boldsymbol{NDe}(X_i)$. $\mathcal{G}$ encodes a set of conditional independence assumptions, known as local independence called $I_l(\mathcal{G})$:*

$$\forall X_i \in \boldsymbol{X} : (X_i \perp\!\!\!\perp \boldsymbol{NDe}(X_i) \mid \boldsymbol{Pa}_{\mathcal{G}}(X_i)) \tag{I.3}$$

The mentioned definition of $I_l(\mathcal{G})$ (I.3) is also called the **Markov assumption**. According to the definition 1.16, a BN structure comprises a set of statements about independence. We aim to demonstrate that a distribution P fulfills the local independence relationships connected with graph $\mathcal{G}$ iff P can be expressed as a group of Conditional Probability Distributions (CPD)s linked with graph $\mathcal{G}$.

Definition 1.17 (Factorization) *Assume that $\mathcal{G}$ is a Bayesian Network structure, and P is a probability distribution over the same set of variables $\boldsymbol{X} = \{X_1, \cdots, X_n\}$. We say that $\mathcal{G}$* factorizes *P as:*

$$P(X_1, \cdots, X_n) = \prod_{i=1}^{n} P(X_i|\boldsymbol{Pa}_{\mathcal{G}}(X_i)) \tag{I.4}$$

The equation I.4 is also known as the Bayesian networks' **chain rule**. The local probabilistic models or Conditional Probability Distributions (**CPD**s) are each of the $P(X_i|\mathbf{Pa}_{\mathcal{G}}(X_i))$ factors that contribute to the overall equation. By considering these factors, we can better understand the likelihood of various outcomes within the network.

Definition 1.18 (Independence Set) *Given the probability distribution P over the set of variables $\boldsymbol{X}$, the set of independence assertions ($\boldsymbol{X}_i \perp\!\!\!\perp_P \boldsymbol{X}_j|\boldsymbol{X}_k$) encoded in P are denoted as $I(P)$, which is the* independence set *of P.*

Definition 1.19 (Active Path) *Given a BN structure $\mathcal{G}$, $\mathbf{Z}$ is a subset of observed variables, the path $X_1 \rightleftharpoons \cdots \rightleftharpoons X_n$ is an* active path *given $\mathbf{Z}$ if for any collider ($X_{i-1} \longrightarrow X_i \longleftarrow X_{i+1}$) in the path, either X_i or one of its descendent in $\mathcal{G}$ are a member of $\mathbf{Z}$, and no other variables in the path is in $\mathbf{Z}$ [Geiger and Pearl 1990]*

To comprehend the concept of probabilistic influence, it can be helpful to visualize it as a flow in a graph. By doing so, we can determine the likelihood of influence coming from A, passing through C, and ultimately impacting our beliefs about B. When it is possible that an influence move from A to B via C, we refer to this path as an **active path**; otherwise, we say the path is **blocked**.

When multiple paths exist between two nodes in a directed graph, assessing whether these nodes are conditionally independent can be crucial. The concept of **d-separation** helps in identifying whether any set of nodes (observed variables) blocks all paths between two nodes, thereby indicating their conditional independence under the given observations.

Definition 1.20 (d-separation) *Let $\mathbf{Z}$ be a set of nodes disjoint from $\{X, Y\}$. X and Y are said d-separated by $\mathbf{Z}$, which is denoted by $\langle X \perp_{\mathcal{G}} Y|\mathbf{Z}\rangle$, if, for every path $\mathscr{C}$ between X and Y, there exists a node $X_i \in \mathscr{C}$ such that one of the following two conditions holds:*

1. *$\langle X_{i-1}, X_i, X_{i+1}\rangle$ is a collider, i.e., $\mathcal{G}$ contains both Arcs $X_{i-1} \rightarrow X_i$ and $X_i \leftarrow X_{i+1}$. In addition, neither X_i nor its descendants in $\mathcal{G}$ belong to $\mathbf{Z}$. The descendants of a node are defined recursively as the union of its children and the descendants of the children.*
2. *$\langle X_{i-1}, X_i, X_{i+1}\rangle$ is not a collider and X_i belongs to $\mathbf{Z}$.*

Let $\mathbf{U}, \mathbf{V}, \mathbf{Z}$ be disjoint sets of nodes. Then $\mathbf{U}$ and $\mathbf{V}$ are d-separated by $\mathbf{Z}$ if and only if X and Y are d-separated by $\mathbf{Z}$ for all $X \in \mathbf{U}$ and all $Y \in \mathbf{V}$ [Pearl 1988].

Let $\mathbf{U}, \mathbf{V}, \mathbf{Z} \subseteq \mathbf{X}$ be disjoint sets and let P be a probability distribution over $\mathbf{X}$. We denote by $\mathbf{U} \perp\!\!\!\perp_P \mathbf{V}|\mathbf{Z}$ the probabilistic conditional independence of $\mathbf{U}$ and $\mathbf{V}$ given $\mathbf{Z}$. The independence model of BNs is the following:

$$\langle \mathbf{U} \perp_{\mathcal{G}} \mathbf{V}|\mathbf{Z}\rangle \Longrightarrow \mathbf{U} \perp\!\!\!\perp_P \mathbf{V}|\mathbf{Z} \tag{I.5}$$

Any graphical model that satisfies Eq. (I.5) is called an I-map (*independence map*) as it is presented in Definition 1.23.

Definition 1.21 (I-equivalence) *The graphs $\mathcal{G}_1$ and, $\mathcal{G}_2$ over the set of variables* **X** *are* I-equivalent *if $I(\mathcal{G}_1) = I(\mathcal{G}_2)$. For every graph over* **X**, *there are sets of I-equivalent graphs with the same independence over* **X**. *These sets are called* I-equivalence classes, *which are mutually exclusive and exhaustive.*

In Definition 1.21, $I(\mathcal{G})$ represents the collection of conditional independence relationships inferred from the network structure:

$$I(\mathcal{G}) = \{\langle \mathbf{U} \perp_{\mathcal{G}} \mathbf{V}|\mathbf{Z}\rangle : d\text{-}sep_{\mathcal{G}}(\mathbf{U};\mathbf{V} \mid \mathbf{Z})\} \tag{I.6}$$

This concept allows us to focus on the relationships between variables rather than the specific details of the graphical representation. It is particularly useful because it reveals that networks with different structures can still represent the same conditional independence assertions.

Theorem 1.1 *Given the graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ over the set of variables* **X**, *if they have the same skeleton and the same set of v-structures, then they are I-equivalent. [Verma and Pearl 1991]*

In theorem 1.1, v-structures refer to the following definition.

Definition 1.22 (v-structure) *Given a graph $\mathcal{G}$, the triple (X, Z, Y) is a* v-structure *if i) edges X —Z, and Z —Y belong to the skeleton and ii) there exist sets $\mathbf{Z} \not\supseteq \{Z\}$ such that $X \perp\!\!\!\perp_P Y|\mathbf{Z}$.*

Definition 1.23 (I-map) *Given a graph $\mathcal{G}$ and a probability distribution P associated with the set of independence assertion $I(\mathcal{G})$ and $I(P)$, respectively, we say that $\mathcal{G}$ is an* I-map *for P if $I(\mathcal{G}) \subseteq I(P)$. If removing any edge from $\mathcal{G}$ makes it not an I-map anymore, then $\mathcal{G}$ is a* minimal I-map.

Definition 1.24 (P-map) *Given a graph $\mathcal{G}$ and a probability distribution P associated with the set of independence assertions $I(\mathcal{G})$ and $I(P)$, respectively, we say that $\mathcal{G}$ is a* perfect map (P-map) *for P if $I(\mathcal{G}) = I(P)$.*

Theorem 1.2 *Given a Bayesian Network structure $\mathcal{G}$, and a probability distribution P over the same set of variables* **X**, *if $\mathcal{G}$ is an I-map of P, then $\mathcal{G}$ factorizes P. [Geiger and Pearl 1990]*

Theorem 1.3 (d-separation soundness) *Given a Bayesian Network structure $\mathcal{G}$, and a probability distribution P over the same set of variables $\mathbf{X}$, if $\mathcal{G}$ factorizes P, then $\mathcal{G}$ is an I-map of P. [Geiger and Pearl 1990]*

To put it simply, when two nodes are d-separated it asserts that they are conditionally independent. D-separation also has the property of completeness, which means that it can capture all the independencies that exist. In Eq. (I.5) if, in addition, the implication is substituted by an equivalence, it is called a P-map (*perfect map*) as it is shown in Definition 1.24. For example, if X and Y are independent given $\mathbf{Z}$, then they are d-separated. But we have to be careful because the property of completeness is not well-defined without specifying the distribution where X and Y are independent. To make this property more precise, we need to introduce the following concept in Definition 1.25

Definition 1.25 (Faithfulness) *Let $\mathcal{G}$ be a Bayesian Network structure and P a probability distribution over the same set of variables. P is* faithful *to $\mathcal{G}$ if, whenever P asserts that X and Y are independent given $\mathbf{Z}$, then they are d-separated in $\mathcal{G}$:*

$$(X \perp\!\!\!\perp_P Y|\mathbf{Z}) \Longrightarrow d\text{-}sep_{\mathcal{G}}(X;Y \mid \mathbf{Z})$$

It means any independence within the distribution P can be seen in the d-separation characteristics of the graph.

The following standard theorem [1] highlights the completeness property of d-separation [Pearl 1988]:

Theorem 1.4 (d-separation Completeness) *Consider $\mathcal{G}$, a graph that factorizes the distribution P, and assume that P is faithful to $\mathcal{G}$. If X and Y are conditionally independent given $\mathbf{Z}$ under P, then X and Y are d-separated by $\mathbf{Z}$ in $\mathcal{G}$.*

Definition 1.26 (Bayesian Network) *A* Bayesian Network *is usually defined as a pair $\mathcal{B} = (\mathcal{G}, P)$, where P is a set of CPDs that correspond to nodes in $\mathcal{G}$, and $\mathcal{G} = (\mathbf{X}, \mathcal{E})$ is the BN structure which factorizes P. And, $\mathbf{X}$ represents a set of random variables, and $\mathcal{E}$ is the set of arcs [Pearl 1988].*

Figure I.2 shows an example of Bayesian Networks over the variables $\{V, W, X, Y, Z\}$. We can calculate the joint probability distribution for this network by factorizing P by the graph:

$$P(V, W, X, Y, Z) = P(X)P(Y)P(Z|X, Y)P(W|Y)P(V|Z)$$

[1] Note that *d-separation Completeness* is a known result from the literature and included here for completeness and clarity, not as a contribution of this thesis.

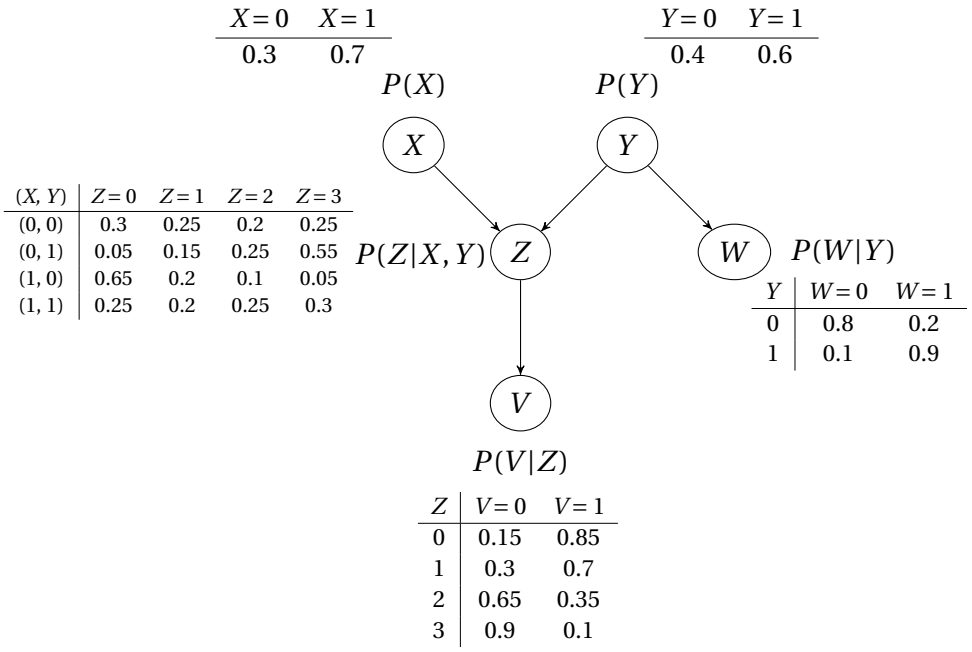


Figure I.2.: A Bayesian Network with CPDs

2. Causal Networks

Causal networks can be considered a particular type of Bayesian network. These models differ from Bayesian networks in that the information they encode accounts for more than just statistical dependencies between variables that we discussed in the previous section. In a causal network, when variable X is the parent of variable Y, it means that X directly causes Y. Put simply, a change in X will result in a change in Y. In this section, first, we will focus on the differences in the representation of Causal Networks compared to the Bayesian Networks (Sections 2.1, and 2.2). Then, we will discuss how to infer a causal query (Section 2.3). Finally, we will talk about different causal learning approaches (Section 2.4).

2.1. Essential Concepts

Some principal categorizations exist in the study of causal and Bayesian networks that differ them, which a reader should be aware of. In the following, we will discuss some of these categorizations that recur more often throughout the current dissertation, which makes it more necessary to explain them in more detail before continuing to the other topics.

2.1.1. Correlation vs. Causation

In the study of causal inference, there is a common and significant concept that correlation does not necessarily imply causation. For example, in machine learning, you may hear that the predicted result does not infer causation. In this section, we will briefly overview this topic using an example.

Figure I.3 by Vigen 2015 expresses that the money spent on science, space, and technology in the US is closely correlated (99.7%) with the rate of suicide by hanging, strangulation, and suffocation. Can we infer that when the rate of suicide changes, it causes the US government to change its decision on how to spend money on space research? (Figure I.4a) Or do people kill themselves because of the US policy on technology investment? (Figure I.4b)

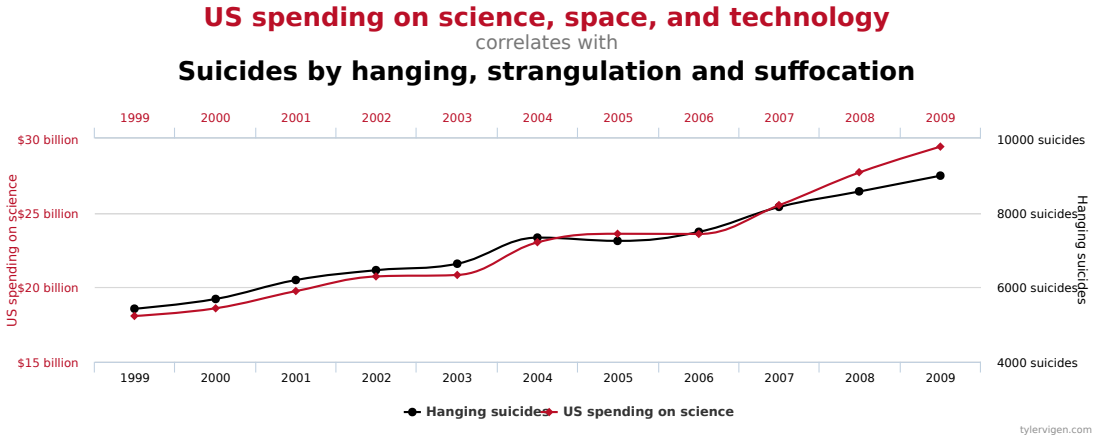
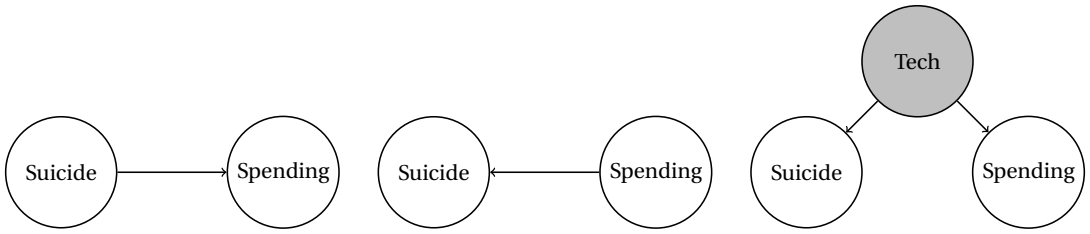


Figure I.3.: Spurious Correlation

It is almost certain that none of these variables is causing the other one, and it is just a spurious correlation. When two unrelated things are tied together, this can be either bound by causality or correlation. In most cases, correlations are just because of the coincidences. Just because it seems one factor is influencing the other, it does not mean it actually does. In some cases, some hidden factors cause two correlated variables. In the mentioned example, it is possible that improvements in technology made people more isolated and less social, and they feel more depressed; hence, they commit suicide. On the other hand, technological improvements make people realize that they can discover even more with more funding. So, a common hidden factor (Improvements in Technology) may cause these two variables to become correlated. (Figure I.5c)

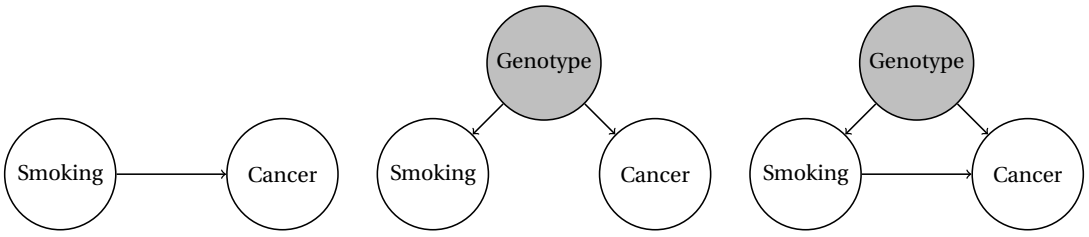


(a) False Scenario 1 (b) False Scenario 2 (c) True Scenario

Figure I.4.: Correlation is not Causation

In the mentioned example, it was pretty evident that the two variables are not causing each other, but what if two variables look like causing each other? For instance, Koller and Friedman 2009 reviewed a case study in 1960 in which scientists found that there is a correlation between smoking and lung cancer. They thought these variables had a causal relationship (Figure I.5a). As a result, they suggested banning smoking, which was against the tobacco industry, and they argued this idea with distinguished

statistics. They claimed that there is another explanation for this correlation: a hidden factor (genotype) causes lung cancer and creates an intrinsic willingness for nicotine (Figure I.5b). These two different structures represent precisely the same distribution over the observable data and are not distinguishable. It is also possible that both of them are right, which means not only do they have a hidden common cause, but there is also a direct causal relationship between them (Figure I.5c), which is not distinguishable from the observational data.



(a) Direct Cause (b) Common Cause (c) Both Types

Figure I.5.: Different Models to Check if Smoking Cause Cancer

The mentioned examples show the importance of understanding the difference between correlation and causation, and later in this part, we will discuss how we can distinguish between different relationships.

2.2. Causal Models

A **causal graph** is a visual representation of a system's cause-and-effect relationships (*Causal Mechanism*). [Bareinboim, Correa, Ibeling, et al. 2022]. In such a mechanism, a variable (X) is causing another variable (Y) if changes in X can cause changes in Y. We can formalize it using the following definition.

Definition 2.1 (Causal Edge Assumption) *In a causal graph, any parent is the direct cause of its children. It means there is a function that maps the values of the parent to the values of children:*

$$X_i := f(\boldsymbol{Pa}(X_i)), \quad \forall X_i \in \boldsymbol{X} \tag{I.7}$$

Equation I.7 is for the ideal case when we know all the parents of the variables, but in the real problems, there are some unobservable (latent) parents:

$$X_i := f(\mathbf{Pa}(X_i), \mathbf{L}(Xi)), \quad \forall X_i \in \mathbf{X} \tag{I.8}$$

Where $\mathbf{L}$ represents the set of unobserved parents.

According to Definition 2.1, when there is a directed edge (arc) from X to Y, X is called the **direct cause** of Y, and Y is the **direct effect**, and any direct cause of the X is denoted as the **indirect cause** of Y. Considering these explanations, we can introduce some principal definitions as follows (Mainly from Pearl 2009).

Definition 2.2 (Functional Causal Model) *A* Functional Causal Model *is a set of equations* $\{X_i := f_i(\mathbf{Pa}(X_i), \mathbf{L}(Xi))\}$, *where* f_i *is a function assigned to* X_i, $\mathbf{Pa}$ *is a set of variables that directly affect* X_i, *and* $\mathbf{L}$ *represents the set of disturbances made because of the unobserved factors.*

Definition 2.3 (Causal Diagram) *Given a functional causal model, if we create a graph in which there is an arc from any members of* $\mathbf{Pa}(X_i)$ *to* X_i, *then this graph is called a* causal diagram.

Definition 2.4 (Semi-Markovian Models) *If a causal diagram is acyclic, and none of the unobserved variables is a descendant of the observed variables, then this graph is called a* Semi-Markovian model. *[Tian and Pearl 2002]*

Tian and Pearl [Tian and Pearl 2015] established that causal models containing arbitrary latent variables can be systematically converted into semi-Markovian causal models (SMCMs). In these transformed models, each latent variable is structured to have no parents and only two children, ensuring that the independence relations among observed variables remain intact

Definition 2.5 (Causal Structure) *The DAG* $\mathcal{G} = (\mathbf{X}, \mathcal{E})$ *is a* causal structure *over a set of variables* $\mathbf{X}$, *where each node represents a unique element of* $\mathbf{X}$. $\mathcal{E}$ *is a set of directed edges (arcs), and each directed edge* ($E \in \mathcal{E}$) *denotes a direct functional relationship between the variables of its endpoints, where the tail is the* cause *and the head is the* effect.

Definition 2.6 (Causal Model) *A* causal model *is a pair* $\mathcal{C} = (\mathcal{G}, \Theta)$, *where* $\mathcal{G} = (\mathbf{X}, \mathcal{E})$ *is a causal structure,* $\mathcal{E}$ *is a set of directed edges (arcs), and a random disturbance* ξ_i *is assigned to each* $X_i \in \mathbf{X}$. $\Theta = \{f_i(\mathbf{Pa}_{\mathcal{G}}(X_i), \xi_i)\}_{X_i \in \mathbf{X}} \bigcup \{P(\xi_i)\}_{X_i \in \mathbf{X}}$, *where the* f_i *is a function assigned to* X_i, *and* P *is the probability over the random disturbance* ξ_i.

Definition 2.7 (Markovian Model) *In a causal model (Definition 2.6), since the error terms* ($\{P(\xi_i)\}_{X_i \in \mathbf{X}}$) *are jointly independent, this model is denoted as a* Markovian model.

2.2.1. Latent Variables

As we discussed in Section 2.1.1, if two variables A and B are both influenced by a common cause C, they are typically correlated. When this common cause C is observed, we can explicitly model its effect and disentangle its influence from the correlation

between A and B. However, in many real-world situations, several factors influencing our variables of interest may not be observed or even identified. Such hidden factors, known as **latent variables**, can lead to observed correlations between variables that do not directly cause each other. Identifying and correctly modeling latent variables thus becomes crucial yet challenging in causal inference from observational data.

The problem of learning latent structures in Bayesian networks has been explored through various frameworks. Historically, early efforts such as the Discrete Factor Analysis proposed by Martin and VanLehn 1995 already considered algorithms aimed at uncovering hidden variables within Bayesian networks using score-based criteria. These pioneering contributions laid foundational hypotheses and methods that closely align with current approaches, including our proposed methods, by using scoring functions to guide the identification of latent structures.

Pearl 2009 further emphasized that multiple Bayesian network structures can be consistent with the same observational data, particularly when latent variables are involved. However, not all structures provide meaningful or practical interpretations. For instance, overly complex models such as those positing a single latent variable causing all observed variables, or overly dense models proposing extensive and unnecessary direct connections, tend to be unrealistic and uninformative. To address this, researchers typically adopt the principle of model simplicity—selecting the simplest plausible structure that adequately fits the observed data. Such structures are termed **minimal** models.

Moreover, it is worth highlighting that our work primarily focuses on observational data contexts. Other notable causal inference methodologies incorporate interventional data or hybrid observational-interventional strategies. Our contribution is thus positioned specifically within the rich literature addressing purely observational contexts, building upon and extending earlier works such as Martin and VanLehn 1995, and offering methods specifically tailored for identifying latent confounders effectively in large-scale observational datasets.

Suppose there is a direct path from two variables in every minimal structure **consistent** with data. In that case, there is a causal influence from the source variable to the destination one. This **inferred causation** is true when all variables are observable. To generalize this definition for structures with latent variables, we present the following definitions mainly from Pearl 2009 (otherwise, the references are noted).

Definition 2.8 (Latent Structure) *The pair $\mathscr{L} = (\mathscr{G}, \mathbf{X}_O)$ is a* latent structure, *where $\mathscr{G}$ is a causal structure over a set variables $\mathbf{X}$, and $\mathbf{X}_O \subseteq \mathbf{X}$ is the set of observed variables.*

Definition 2.9 (Independence Set (Latent Structure)) *Let $\mathscr{L} = (\mathscr{G}, \mathbf{X}_O)$ be a latent structure. $I_{\mathbf{X}_O}(\mathscr{L})$ is the* independence set *of $\mathscr{L}$ which includes $(\mathbf{W} \perp \mathbf{Y} | \mathbf{Z}) \in I_{\mathbf{X}_O}(\mathscr{L})$ for $\mathbf{W}, \mathbf{Y}, \mathbf{Z} \subset \mathbf{X}_O$ [Koller and Friedman 2009]*

Definition 2.10 (I-equivalence (Latent Structure)) *Let $\mathscr{L}_1$ and $\mathscr{L}_2$ be two latent structures over a set of observable variables $\mathbf{X}_O$. $\mathscr{L}_1$ and $\mathscr{L}_2$ are in the same class and called $I_{\mathbf{X}_O}$-equivalent if $I_{\mathbf{X}_O}(\mathscr{L}_1) = I_{\mathbf{X}_O}(\mathscr{L}_2)$ [Koller and Friedman 2009]*

Definition 2.11 (Structure Preference) *The latent structure $\mathscr{L} = (\mathscr{G}, \mathbf{X}_O)$ is* preferred *to $\mathscr{L}' = (\mathscr{G}', \mathbf{X}_O)$ (written $\mathscr{L} \preceq \mathscr{L}'$) iff $\mathscr{G}'$ can act like $\mathscr{G}$ over $\mathbf{X}_O$, i.e., iff there exists a Θ' for every Θ s.t. $P_{\Theta'}(\mathbf{X}_O) = P_{\Theta}(\mathbf{X}_O)$ over observed variables, where $\mathscr{C}$ and $\mathscr{C}'$ are causal models with different parameters (Θ and Θ' resp.). Two latent structures are equivalent, written $\mathscr{L} \equiv \mathscr{L}'$, iff $\mathscr{L}$ is preferred to $\mathscr{L}'$ ($\mathscr{L} \preceq \mathscr{L}'$) and $\mathscr{L}'$ is preferred to $\mathscr{L}$ ($\mathscr{L}' \preceq \mathscr{L}$).*

Definition 2.11 represents the preference among different models that have latent variables to help choose the model that is simpler, which means that the model does not have many parts that are not needed to explain the data. Generally, scientists like simple models because they are easier to test and trust and less prone to overfit the data, i.e. if a simple model fits the data well, it is more convincing than a complex model. The following definition talks about the most preferred model.

Definition 2.12 (Minimality) *Among the members of a class $\mathbf{L}$ of latent structures, a structure $\mathscr{L}$ is* minimal *iff it is equally or more preferred than any other member and there is no member strictly preferred to $\mathscr{L}$. In other words, $\mathscr{L}$ is minimal iff, for any other structure $\mathscr{L}'$ in $\mathbf{L}$, $\mathscr{L}' \preceq \mathscr{L}$ implies $\mathscr{L} \equiv \mathscr{L}'$.*

Definition 2.13 (Consistency) *A latent structure $\mathscr{L} = (\mathscr{G}, \mathbf{X}_O)$ is consistent with a distribution P^* over $\mathbf{X}_O$, if there exists a parameterization Θ such that the induced probability distribution $P_\Theta(\mathbf{X}_O)$ from the causal model $(\mathscr{G}, \Theta)$ matches the distribution P^*.*

In other words, for a model $\mathscr{L}$ to be consistent with a distribution P^*, $\mathscr{L}$ must be able to explain all the dependencies between the variables in P^*. For example, if P^* encodes that A and B are dependent, then $\mathscr{L}$ must have a causal connection between A and B. If $\mathscr{L}$ cannot explain some of the dependencies in P^*, then $\mathscr{L}$ is inconsistent with P^*. This is a necessary condition, i.e., it must be true for $\mathscr{L}$ to be consistent, but it is not always a sufficient condition, meaning that it is not enough by itself. Sometimes, $\mathscr{L}$ must also satisfy other conditions, such as numerical constraints or simplicity criteria, to be consistent with P. [Verma and Pearl 1991] Now, we can present the generalized definition of the inferred causation in the presence of the latent variables.

Definition 2.14 (Inferred Causation) *Given a distribution P', a variable X influences another variable Y causally, iff in every minimal latent structure consistent with P', a directed path exists from X to Y.*

We reviewed the abovementioned concepts to find the simplest model, but it is possible that the simplest model is not the true model or that many simple models

have different structures while they are not distinguishable from their parameters. For example, imagine you have three variables that are related to each other by flipping two coins. One variable is the result of the first coin (X), another is the result of the second coin (Y), and the third is whether the two are the same (Z). The data from these variables can be explained by three different models, each saying that one variable is caused by the other two. But there is no way to tell which one is the true model from the data. To avoid this problem, another assumption is needed. This assumption is called **stability**, or DAG-isomorphism and perfect-mapness [Pearl 1988], or faithfulness [Peter Spirtes, C. N. Glymour, and Richard Scheines 2000]. It says that the model matches the data precisely and that the data shows all the independencies in the model, and they are stable even if the parameters (Θ) change. In the mentioned example, only the true model ($X \longrightarrow Z \longleftarrow Y$) will keep its dependencies when we change the parameters (coins settings, e.g., if we make the coins more or less fair).

Definition 2.15 (Stability) *Given the independence set $I(P)$ (Definition 1.18), a causal model $\mathscr{C} = (\mathscr{G}, \Theta)$ has a stable distribution iff $I(P(\mathscr{C})) \subseteq I(P(\mathscr{C}'))$ for any Θ', where $\mathscr{C}' = (\mathscr{G}, \Theta')$.*

But when some variables are not observed, stability is insufficient. It means there is no longer any guarantee of finding a true DAG structure. Pearl 2009 showed that it is possible to solve this problem by limiting the search space of the latent structures to some specific structures. He expressed that any latent structure has a simpler equivalent structure that shows the same dependencies. This structure has only latent variables at the top, each with exactly two observable variables as children. He called these structures the **projection** of the original structures.

Definition 2.16 (Projection) *$\mathscr{L}' = (\mathscr{G}', \mathbf{X}_O)$ is a projection of $\mathscr{L} = (\mathscr{G}, \mathbf{X}_O)$ iff: I. Every latent variable in $\mathscr{G}$ has no parents, and each causes exactly two observable variables that are not connected. II. For every stable distribution P that $\mathscr{L}$ encodes, there is a stable distribution P' that $\mathscr{L}'$ encodes s.t. $I(P) = I(P')$ where $I(.)$ refers to the independence set (Definition 1.18).*

2.2.2. Partially Directed Acyclic Graphs

We have seen in the previous sections how the causal edge assumption leads to the idea of the causal graph, where the parents directly cause their children. However, a simple DAG does not show how much we know or do not know about the causal relationship among the variables. So, we require a formal definition for other graphical structures that explain the causes and effects, which can be satisfied by **PDAG**s and **CPDAG**s [Peter Spirtes, C. N. Glymour, and Richard Scheines 2000, David Maxwell Chickering 2002a].

Definition 2.17 (Partially Directed Acyclic Graph (PDAG)) *A* PDAG *is a graph that contains both directed and undirected edges, and it has no directed cycles.*

This representation of the graphical structures differentiates a causal relationship ($X \longrightarrow Y$) from an unknown relationship (X—Y), where the edge direction is uncertain. This discrimination helps to represent a new definition of equivalency of different networks.

Definition 2.18 (Observational Equivalence) *Two DAG structures $\mathcal{G}_1$ and $\mathcal{G}_2$ are observational Markov equivalent if they have the same skeleton (Definition 1.2) and the same set of v-structures (Definition 1.22) and written as $\mathcal{G}_1 \cong \mathcal{G}_2$ [Verma and Pearl 1991]*

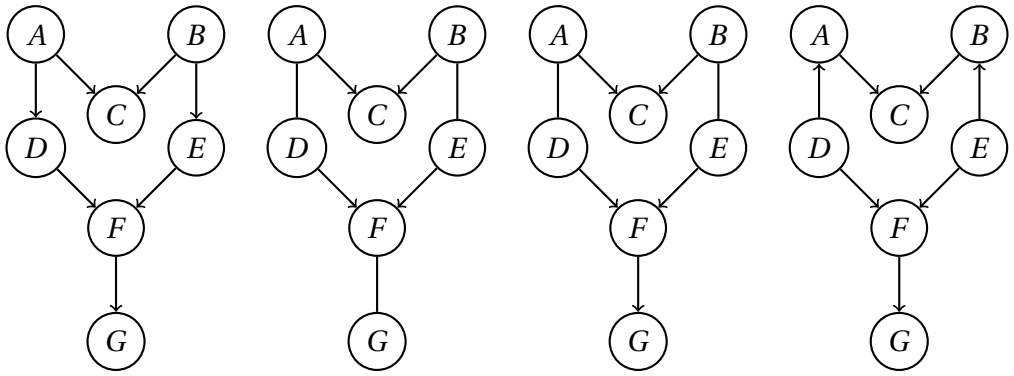
Definition 2.19 (Observational Equivalence Class) *Two DAG structures $\mathcal{G}_1$ and $\mathcal{G}_2$ are in the same observational Markov equivalence class (MEC) if they are observational equivalent. Broadly speaking, the observational MEC of graph $\mathcal{G}$, written as $[\mathcal{G}]$, corresponds to the group of all DAGs that are observational equivalent. [K. Yang, Katcoff, and Uhler 2018, J. M. Mooij and Claassen 2020, J. M. Mooij, Magliacane, and Claassen 2020]*

In fact, these definitions are obtained from the theorem 1.1. The v-structures in Definition 2.18 are essential because they show the conditional independencies that affect the related probability distribution. An arc that would make a different v-structure if reversed is called a **compelled arc**, as shown in Figure I.6b where we can see the v-structures of the DAG (Figure I.6a). An arc that is not compelled is called *reversible* and can be converted to an (undirected) edge to have a causal graph.

Definition 2.20 (Compelled Arc) *An arc ($\longrightarrow$) is* compelled *if it exists in all I-equivalent graphs (every member of the I-equivalence class).*

If the directed arcs in a graph only include the compelled arcs, and there are (undirected) edges for the rest of the arcs of the initial DAG, then we would have the completed PDAG (**CPDAG**) of the DAG as shown in Figure I.6c. The directed arcs in this figure are compelled because reversing them would create a new v-structure. A compelled arc, together with the basic skeleton, limits the range of observational distributions that match the given DAG.

Definition 2.21 (Completed Partially Directed Acyclic Graph (CPDAG)) *A* CPDAG *is a completed PDAG corresponding to an I-equivalence class, i.e., it has an arc (resp. edge) for all compelled (resp. reversible) arcs of the DAGs in the MEC.*



(a) A DAG $\mathcal{G}$ (b) A PDAG of $\mathcal{G}$ (c) The CPDAG of $\mathcal{G}$ (d) An MEC of $\mathcal{G}$

Figure I.6.: DAG vs. Partially DAGs

Definitions 2.17 and 2.21 may be referred to with different names in the literature. For instance, PDAGs are called *patterns* in Peter Spirtes, Clark Glymour, Richard Scheines, et al. 1993 and Peter Spirtes, C. N. Glymour, and Richard Scheines 2000, and CPDAGs are mentioned as *essential graphs* by Andersson, Madigan, and Perlman 1997 and *maximally oriented graphs* by Meek 1995.

Definition 2.20 represents that an arc is compelled when we cannot reverse it; the other connections are reversible, and that is the reason they exist as an (undirected) edge; therefore, in CPDAGs, the (directed) arcs represent the invariant causal relationships across all DAGs in the equivalence class. In contrast, the (undirected) edges represent ambiguous or reversible relationships in a causal sense.

2.3. Causal Inference

Causal inference involves determining the impact of one variable on another. This can be used to evaluate the effectiveness of treatments or interventions. For instance, if you observe a rise in some product sales and suspect that the new advertising method contributes, you can conduct a causal analysis to quantify its impact. Overall, causal inference is valuable for examining the relationship between any variable X and its effect on another Y.

2.3.1. Observational vs. Interventional

Finding out the causal effects among variables is much easier when we can do experiments or *interventions*. The reason is that we can perform our desired action and observe the outcome of it. It is more difficult when we only have *observational* data because there are usually other factors, such as (hidden) common causes, that can influence our causal inferences.

Understanding the difference between observational and interventional data is very important in causal approaches. It can be explained by the causal hierarchy, also known as the ladder of causation [Pearl and Mackenzie 2018, Bareinboim, Correa, Ibeling, et al. 2022, Zanga, Ozkirimli, and Stella 2022]. This hierarchy has three layers corresponding to the level of cognition: observational, interventional, and counterfactual. The observational layer which has the lowest cognition level, represents the factual data and association between variables, whereas the interventional layer deals with the data generated through experiments that affect the values variables can take. The counterfactual layer has the highest cognition level since it corresponds to situations when the desired outcome is for different interventions than what is actually available among the variables.

In their foundational work, Pearl and colleagues [Pearl 1994, Pearl 2009] introduced the do-calculus, which provides a formal framework for the analysis of causal effects within the causal hierarchy (Definition 2.24). The do-calculus utilizes the $do(.)$ operator to represent hypothetical interventions within a causal model. When we apply an intervention $do(\mathbf{X} = \mathbf{x})$ on a set of variables $\mathbf{X}$, we are essentially simulating the act of setting $\mathbf{X}$ to the values $\mathbf{x}$ in a controlled manner, independent of their usual causal influences. This manipulation is conceptually akin to performing a controlled experiment where $\mathbf{X}$ is no longer determined by its prior causes within the system.

The impact of such an intervention on another set of variables $\mathbf{Y}$ is expressed using the interventional distribution $P(Y \mid do(\mathbf{X}))$, which captures the causal effect of the intervention on $\mathbf{Y}$. Graphically, this is represented by modifying the original causal structure $\mathcal{G}$: for a variable X subjected to an intervention, we remove all incoming arcs to X, resulting in a modified graph denoted as $\mathcal{G}_{\overline{\mathbf{X}}}$. If we intervene on all children of a variable $\mathbf{Z}$, we remove all outgoing arcs from $\mathbf{Z}$, leading to a graph represented as $\mathcal{G}_{\underline{\mathbf{Z}}}$. These graphical alterations do not imply the elimination of the underlying causal mechanisms; rather, they symbolize the isolation of variables from their usual determinants within the model, allowing for the estimation of direct causal effects [Shpitser and Pearl 2008].

It is important to note that the simplicity of these interventions in the graphical model does not always fully account for the complexity of real-world causal mechanisms. Not all causes may be removed by the intervention—some may persist or be unobserved—highlighting the difference between the idealized manipulations in the model and the practical limitations of interventions in practice. Zanga, Ozkirimli, and Stella 2022 categorized the interventions with the following definitions based on their perfectness, i.e., whether they remove the causal mechanism.

Definition 2.22 (Perfect Intervention) *An intervention is called* perfect *when it removes the causal dependencies between the target and its causes, i.e., removes the incoming causal arcs in the causal structure.*

We will discuss these kinds of interventions in the following Section, where we introduce Pearl 2009's do-calculus since they consider the intervention perfect. However, other interventions exist that are not perfect, as shown in the following definition.

Definition 2.23 (Imperfect Intervention) *An intervention is called* imperfect *if it does not remove the causal dependencies between the target and its causes, and in contrast, it changes the causal function between them.*

Perfect and Imperfect interventions, also called by other names in the literature, for instance, Markowetz, Grossmann, and Spang 2005 denoted them as *hard* and *soft* interventions, respectively. The imperfect intervention is actually part of a more general concept known as *mechanism change* proposed by Tian and Pearl 2013, as per its formal definition.

2.3.2. do-calculus and Identifiability

As we can see in the definitions of previous sections, causal models are associated with a probability distribution, which led Pearl 1994 to propose some rules for making inferences from causal models. These rules include the usual principles of probability, besides some others that deal with counterfactuals, such as effectiveness, composition, and so on [Galles and Pearl 1998, Halpern 2000]. The following definition focuses on three useful rules for finding interventional distributions. These rules are called **do-calculus**:

Definition 2.24 (do-calculus) *Given a causal model $\mathscr{C} = (\mathscr{G}, \Theta)$ which induces a probability distribution P, the* do-calculus *are defined as follows for any disjoint subset of variables $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$, and $\mathbf{W}$: [Pearl 2009]*

1. Insertion and Deletion of Observations:

$$P(\mathbf{Y} \mid do(\mathbf{X}),\ \mathbf{Z}, \mathbf{W}) = P(\mathbf{Y} \mid do(\mathbf{X}), \mathbf{W}) \quad if\ (\mathbf{Y} \perp\!\!\!\perp \mathbf{Z}) | \mathbf{X}, \mathbf{W})_{\mathscr{G}_{\overline{X}}} \tag{I.9}$$

2. Exchange of Observations and Interventions:

$$P(\mathbf{Y} \mid do(\mathbf{X}),\ do(\mathbf{Z}), \mathbf{W}) = P(\mathbf{Y} \mid do(\mathbf{X}),\ \mathbf{Z}, \mathbf{W}) \quad if\ (\mathbf{Y} \perp\!\!\!\perp \mathbf{Z}) | \mathbf{X}, \mathbf{W})_{\mathscr{G}_{\overline{\mathbf{X}},\underline{\mathbf{Z}}}} \tag{I.10}$$

3. Insertion and Deletion of Interventions:

$$P(\mathbf{Y} \mid do(\mathbf{X}),\ do(\mathbf{Z}), \mathbf{W}) = P(\mathbf{Y} \mid do(\mathbf{X}), \mathbf{W}) \quad if\ (\mathbf{Y} \perp\!\!\!\perp \mathbf{Z}) | \mathbf{X}, \mathbf{W})_{\mathscr{G}_{\overline{\mathbf{X}},\overline{\mathbf{Z(W)}}}} \tag{I.11}$$

where $\mathscr{G}_{\overline{\mathbf{X}}}$ (resp. $\mathscr{G}_{\underline{\mathbf{Z}}}$) represents the graph where the incoming (resp. outgoing) arcs to $\mathbf{X}$ (resp. $\mathbf{Z}$) are removed, and $\mathbf{Z(W)}$ indicates the subset of nodes in $\mathbf{Z}$ which are not the ancestors of any nodes in $\mathbf{W}$ in $\mathscr{G}_{\overline{\mathbf{X}}}$. $P(\mathbf{Y} \mid do(\mathbf{X}), \mathbf{Z})$ refers to the probability distribution

of $\mathbf{Y}$ *with the condition that* $\mathbf{Z}$ *is observed and intervention on* $\mathbf{X}$ *to have the constant value* $\mathbf{x}$.

To better understand these definitions, we bring an example. Consider a person (i) who has a disease and has taken some medications as treatment (T) and assume the recovery (Y) as the outcome. There are two possible scenarios. In the first scenario, i gets better ($Y = 1$) after taking the medication ($T = 1$). However, i might also recover without the medication ($T = 0$), so the chance of T causing Y is slight in this case. In the second scenario, the medication is the only cure for i. In this case, the possibility of T causing Y is much higher than in the previous scenario. Suppose we want to use do-calculus to determine the causal effect of medication on the recovery process. In that case, we can compare the probability distribution of $P(Y = 1 \mid do(T = 1))$ and $P(Y = 1 \mid do(T = 0))$.

Some other definitions in the literature try to formalize this problem and help employ do-calculus to estimate the causal effect between the variables using the observable data, and we represent them as follows.

Definition 2.25 (Identifiability) Identifiability *is a property that determines whether a causal query (e.g.,* $P(Y \mid do(X))$*) can be answered using only conditional probabilities over the observable variables. [Koller and Friedman 2009]*

Definition 2.26 (Back-Door Path) *An active path from* X *to* Y *is a* back-door path *if it leaves* X *from its parents toward* Y*, i.e., the path begins with an incoming arc to* X*. [Shpitser and Pearl 2008, Koller and Friedman 2009]*

Definition 2.27 (Back-Door Criterion) *Given a query* $P(\mathbf{Y} \mid do(\mathbf{X}))$ *in a DAG, a set of variables* $\mathbf{B}$ *satisfies the* back-door criterion *if for any* $X \in \mathbf{X}$ *and* $Y \in \mathbf{Y}$*:*

- *no node in* $\mathbf{B}$ *is a descendant of* X
- $\mathbf{B}$ *blocks all the back-door paths from* X *to* Y *[Pearl 1993]*

Definition 2.28 (Back-Door Adjustment) *Let* $\mathbf{B}$ *be a set of variables, if* $\mathbf{B}$ *satisfies the back-door criterion between* $\mathbf{X}$ *and* $\mathbf{Y}$*, then the causal effect of* $\mathbf{X}$ *on* $\mathbf{Y}$ *is identifiable and equals to: [Pearl 2009]*

$$P(\mathbf{Y} \mid do(\mathbf{X})) = \sum_{\mathbf{B}} P(\mathbf{Y} \mid \mathbf{X}, \mathbf{B}) P(\mathbf{B}) \tag{I.12}$$

Definition 2.29 (Front-Door Path) *An active path from* X *to* Y *is a* front-door path *if it leaves* X *from its children toward* Y*, i.e., the path begins with an outgoing arc from* X*. [Shpitser and Pearl 2008]*

Definition 2.30 (Front-Door Criterion) *Given a query* $P(\mathbf{Y} \mid do(\mathbf{X}))$ *in a DAG, a set of variables* $\mathbf{F}$ *satisfies the* front-door criterion *if for any* $X \in \mathbf{X}$ *and* $Y \in \mathbf{Y}$*:*

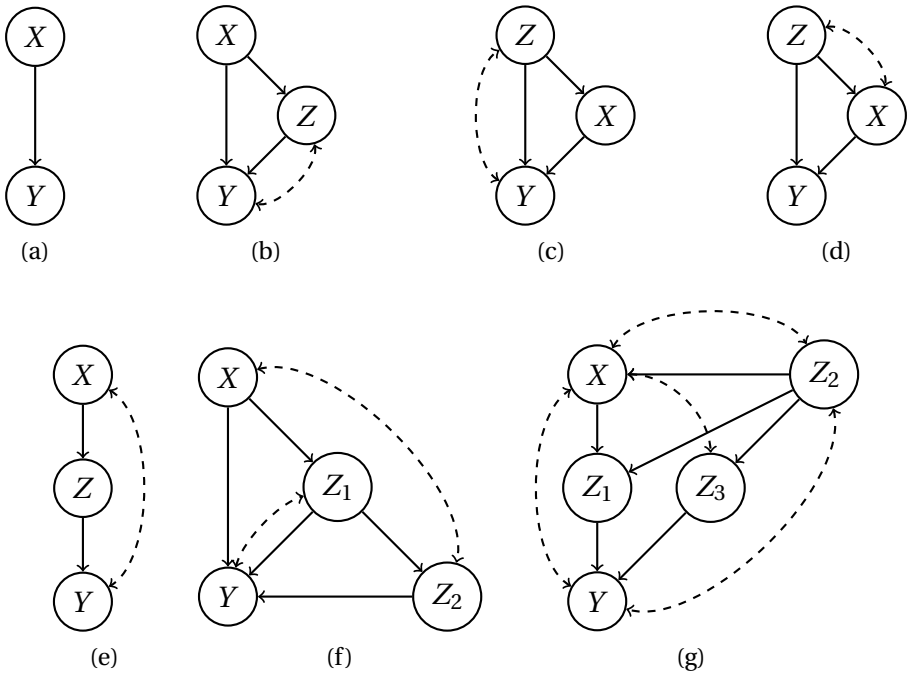


Figure I.7.: Identifiable Causal Effect $P(\mathbf{Y} \mid do(\mathbf{X}))$

- $\mathbf{F}$ *blocks all front-door paths from* X *to* Y
- *There is not any active back-door path from* X *to* $\mathbf{F}$
- *All back-door paths from* $\mathbf{F}$ *to* Y *are blocked by* X *[Pearl 1995]*

Definition 2.31 (Front-Door Adjustment) *Let* $\mathbf{F}$ *be a set of variables, if* $\mathbf{F}$ *satisfies the back-door criterion between* $\mathbf{X}$ *and* $\mathbf{Y}$*, then the causal effect of* $\mathbf{X}$ *on* $\mathbf{Y}$ *is identifiable and equals to: [Pearl 2009]*

$$P(\mathbf{Y} \mid do(\mathbf{X})) = \sum_{\mathbf{F}} P(\mathbf{F} \mid \mathbf{X}) \sum_{\mathbf{X}'} P(\mathbf{Y} \mid \mathbf{X}', \mathbf{F}) P(\mathbf{X}') \tag{I.13}$$

Figures I.7 and I.8 from Shpitser and Pearl 2008 show some graphs as an example for the mentioned definitions to check how we examine identifiability using observational data without needing interventional data. In these graphs, the presence of the latent variables is shown by a dotted curved line which is called **bow arc** (or bow pattern in Koller and Friedman 2009).

Definition 2.32 (Bow Arc) *A bi-directed (dashed) arc between two observable variables is called a* bow arc*, meaning that a latent variable is a common cause of them. [Shpitser and Pearl 2008]*

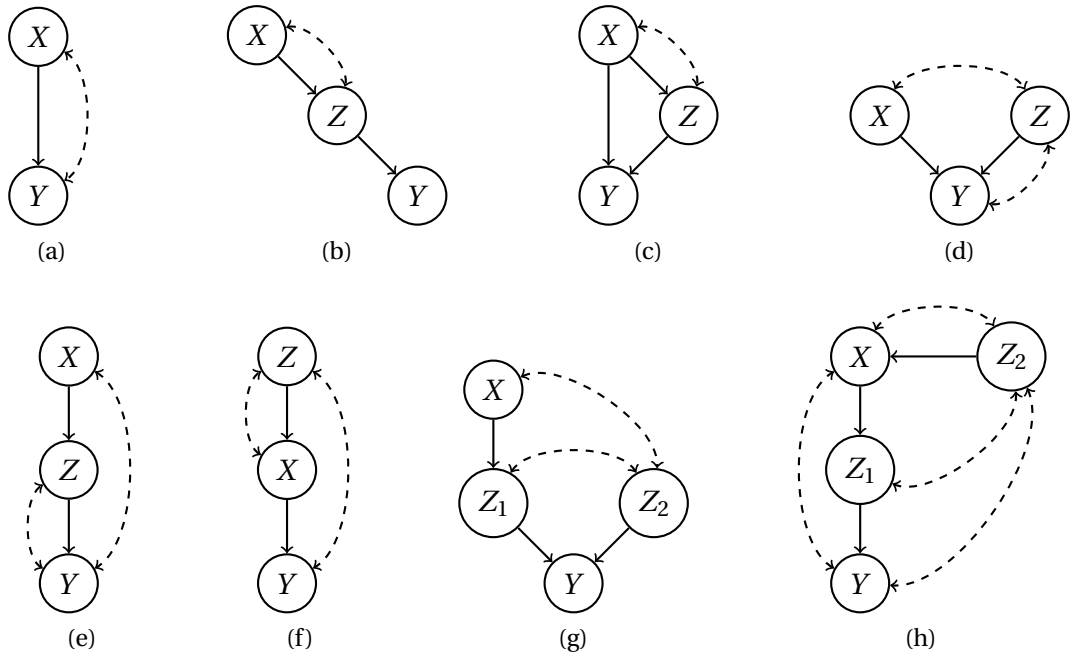


Figure I.8.: Not Identifiable Causal Effect $P(\mathbf{Y} \mid do(\mathbf{X}))$

Shpitser showed that when all d-connecting paths from X to Y are front-door, $P(Y \mid do(X))$ is identifiable, like what we see in Figures I.7a and I.7b where $(Y \mid do(X)) = P(Y|X)$. In Figures I.7c, and I.7d we can see node Z satisfies the back-door criterion and $P(Y \mid do(X))$ is identifiable. Figure I.7e represents the front-door criterion where Z satisfies it, and $P(Y \mid do(X))$ is identifiable.

We can imagine that there might be many cases in which none of the front-door and back-door criteria are satisfied, like the graphs in Figure I.8. As you can see, Figure I.8a shows one of the simplest structures with this property, called the **bow arc graph**. However, it does not necessarily mean the causal effect is not identifiable whenever none of the criteria is satisfied. For example, figures I.7f and I.7g are identifiable even though the back-door and front-door criteria are unsatisfied. To understand why such cases occur, we require more concepts than those mentioned. All of the following definitions and theorems are from Shpitser and Pearl 2008.

Definition 2.33 (Tree) *A graph is called* tree *if any node can have at most one child. In a tree, there is only one node that has no child, called the tree's root.*

All of the graphs in Figure I.8 (disregarding the bow arcs) are trees except Figure I.8c.

Definition 2.34 (C-Path) *A path that connects the observable nodes only using the bow arcs is called a* confounding path *or* C-path.

For instance, in Figure I.8h, there are two C-paths between Z_2 and Y.

Definition 2.35 (C-Component) *A graph is* C-component *if a C-path exists between any pair of observable nodes.*

Definition 2.36 (C-Tree) *A graph is* C-tree *if it is both a C-component and a tree.*

Except for the Figures I.8b, I.8c, I.8g, all the graphs in Figure I.8 are Y-rooted C-trees (i.e., Y is the root of these graphs).

Theorem 2.1 *For any set of nodes* $\mathbf{X} \subset \mathbf{V}$ *in a* Y*-rooted C-tree, where* $\mathbf{V}$ *is the set of all the variables, and* $Y \notin \mathbf{X}$*, Then* $P(Y \mid do(\mathbf{X}))$ *is not identifiable.*

Theorem 2.2 $P(Y \mid do(Pa(Y)))$ *is not identifiable iff there exists a* Y*-rooted C-tree among the subgraphs of the structure.*

As we can see in these theorems, the Y-rooted C-trees play a significant role in identifying the causal effects between different variables, which facilitates differentiation of structures in Figures I.7 and I.8 in terms of identifiability.

2.3.3. Potential Outcomes

The **potential outcome** framework[Rubin 1974], also known as the Rubin Causal Model[Holland 1986], is a famous method for finding the causal effects among variables. In the example that we brought for do-calculus (Definition 2.24), $Y(1)$ and $Y(0)$ represent the *potential outcomes* of recovery when the treatment (T) is set to 1 and 0, respectively. The first scenario assumes that $Y(1)$ and $Y(0)$ are both equal to 1, while the second scenario assumes that $Y(1)$ is equal to 1 and $Y(0)$ is 0. In this example, we considered just one person to examine, whereas, typically, experiments involve multiple individuals, so the potential outcome of an individual is denoted as $Y_i(t)$, where i and t represent the individual and the treatment, respectively. Since different individuals can have different potential outcomes, $Y(t)$ is a random variable. However, $Y_i(t)$ is not a random variable because estimating its value requires conditions on other information that make the potential outcomes deterministic.[Neal 2020]

Definition 2.37 (Potential Outcome (PO)) *The* potential outcome *of* $Y = y$ *for an individual* $X = i$*, when the treatment* $T = t \in \{0,1\}$ *is applied, is denoted as* $Y_i(t)$ *is the value of* $Y(t)$ *for the individual* i *being treated by* t *and it is equal to [Holland 1986] :*

$$Y_i(t) = Y_i(0) + T_i.(Y_i(1) - Y_i(0)) \tag{I.14}$$

Using the potential outcome model, we can estimate different causal effects of interest as follows : [Rubin 2005]

i	T	Y	$Y(1)$	$Y(0)$	$Y(1)-Y(0)$
1	1	1	1	-	-
2	1	0	0	-	-
3	0	0	-	0	-
4	0	1	-	1	-
5	1	1	1	-	-
6	1	1	1	-	-
7	0	0	-	0	-
8	1	1	1	-	-
9	1	0	0	-	-

Table I.1.: Artificial Data for Disease Treatment

Definition 2.38 (Individual Treatment Effect (ITE)) *Each element in the potential outcome model has an* Individual Treatment Effect (ITE) *model that measures the impact of treatment on the individual, which is equal to:*

$$ITE_i(t) = Y_i(1) - Y_i(0) \tag{I.15}$$

Based on these definitions, we can see that it is impossible to see all the potential outcomes for an individual, which makes the estimation of ITE very difficult [Rubin 1974]. For instance, in our mentioned example, you can imagine we cannot examine both treatments on a single patient. And we have the output of a single treatment in the data. This is the *fundamental problem of causal inference* [Holland 1986] since we cannot observe the $Y_i(1)$ and $Y_i(0)$ simultaneously, therefore calculation of the causal effect $Y_i(1) - Y_i(0)$ is not possible. Using a dataset, we have some facts that let us observe a potential outcome. The PO that we can not observe from the data is called the **counterfactual** because it is counter the facts, and conversely, the PO that we can observe is called **factual**. To see why this is a fundamental problem, consider the example on the previous page about a randomized experiment for treating a disease. Imagine Table I.1 represents the data for this example. As can be seen, there are many missing values in this table, and this is exactly the fundamental problem of causal inference, since in our data, either we know the output of having a treatment or not having it, we cannot be aware of both. Hence, the calculation of ITE is not possible based on observational data, and we need another definition that makes the calculations possible. The concept of average treatment effect (ATE) may be helpful to solve this issue. That is the difference between the average outcome of the units receiving the treatment and those that do not.

Definition 2.39 (Average Treatment Effect (ATE)) *In a potential outcome model,* Average Treatment Effect (ATE) *measures the average impact of treatment on the individuals*

(ITEs), which is equal to:

$$ATE(t) = \mathbb{E}[Y_i(1) - Y_i(0)] = \mathbb{E}[Y(1) - Y(0)] \tag{I.16}$$

We need to see if ATE can solve the mentioned missing data problem, which is the idea that we can think of the potential outcomes of each unit as existing, but only one of them is observed, depending on whether the unit receives the treatment. The other potential outcome is missing, but it is not missing at random. It is missing because of the treatment assignment mechanism, which may depend on some factors that affect the outcome. For clarification, consider the example of Table I.1. To calculate ATE based on this table, a quick solution that may come to our mind is that $\mathbb{E}[Y(1) - Y(0)]$ equals $\mathbb{E}[Y|T = 1] - \mathbb{E}[Y|T = 0]$ which can be calculated easily from the observational results in Table I.1, which is not a true answer since this solution is a statistical inference not causal. In the mentioned answer, we ignored the presence of a confounder (Figure I.9b). In contrast, in a causal experiment, the treatment assignment may depend on some confounding variables that affect the potential outcomes (Figure I.9a).



(a) Confounding (b) Non-Confounding (Ignorability)

Figure I.9.: Causal Structures

Therefore, we need an assumption that allows us to identify the ATE from the observed data without requiring an intervention or a randomized experiment. **Ignorability** is such an assumption [Rosenbaum and Rubin 1983, Pratt and Schlaifer 1988] that says the treatment assignment is independent of the potential outcomes, given some observed covariates that account for the confounding factors.

Definition 2.40 (Ignorability) *Assuming that the treatment is independent of the potential outcomes, given the observed covariates (confounders), is called the* Ignorability *assumption and is formalized as follows:*

$$(Y(1), Y(0)) \perp\!\!\!\perp T \mid \boldsymbol{X} \tag{I.17}$$

A weaker version of ignorability called **exchangeability** does not require conditioning on any covariates. It means the treatment assignment is independent of the potential outcomes, regardless of any confounding factors. In other words, it indicates that $\mathbb{E}[Y(1)|T=1] = \mathbb{E}[Y(1)|T=0]$ and $\mathbb{E}[Y(0)|T=1] = \mathbb{E}[Y(0)|T=0]$.

Definition 2.41 (Exchangeability) Exchangeability *is a weaker version of the ignorability assumption, where it removes the condition of the confounding variable [Neal 2020]:*

$$(Y(1), Y(0)) \perp\!\!\!\perp T \tag{I.18}$$

As we can see, ignorability and exchangeability assumptions are fundamental because they are the key to identifying the causal effect of a treatment on an outcome from observational data without the need to conduct an intervention or a randomized experiment. They allow us to use the observed data to estimate the missing data and thus compute the ATE. Without these assumptions, we cannot identify the ATE from the observed data and may need to resort to other methods.

Based on what inference we are looking for, we may need other definitions. For instance, if we care about the average over the treated individuals, we will need the Average Treatment Effect on Treated as represented in Definition 2.42. [Nogueira, Pugnana, Ruggieri, et al. 2022]

Definition 2.42 (Average Treatment Effect on Treated (ATT)) Average Treatment Effect on Treated *equals ATE when it is restricted to the treated individuals:*

$$ATT(t) = \mathbb{E}[Y_i(1) - Y_i(0)|T=1] = \mathbb{E}[Y(1) - Y(0)|T=1] \tag{I.19}$$

The presence of dissimilarity among the individuals may cause the ATE to fail to find the correct causal effects, but this can be solved by using the Conditional Average Treatment Effect, which is shown in Definition 2.43.

Definition 2.43 (Conditional Average Treatment Effect (CATE)) Conditional Average Treatment Effect *equals ATE given a subset of the individuals (***X***):*

$$CATE_i(t) = \mathbb{E}[Y_i(1) - Y_i(0)|\mathbf{X}=\mathbf{x}] = \mathbb{E}[Y(1) - Y(0)|\mathbf{X}=\mathbf{x}] \tag{I.20}$$

Definition 2.43 shows that if $\mathbf{x} = \emptyset$ then CATE = ATE

2.4. Causal Learning

Causal models are useful for understanding how variables affect each other and what happens when we intervene in them. However, learning causal models is not easy, and there are many choices and issues to consider. Some of the main ones are: Do we know the structure of the model (Parameter Learning), or do we have to find it from the data (Structure Learning)? What kind of data do we have (Just observations or also interventions)? Interventions can help us distinguish between different models, but they are often hard to do (even impossible, illegal, or immoral). Do we assume that there are no latent variables that can confuse our causal inferences, or do we account for them? If we ignore them, the problem is easier, but we may miss some important causes. If we include them, the problem is harder, and we may have less certainty.

Peter Spirtes, C. N. Glymour, and Richard Scheines 2000, and Peter Spirtes and K. Zhang 2016 denoted these issues as **causal discovery problem**, which refers to finding a proper explanation for the causal relationship among the variables of a dataset by choosing a suitable graphical structure. (Definition 2.55)

In this section, we will focus on learning causal models that help us to find the causal direction between variables, perform inferences among them, and answer interventional questions. In other words, we try to see how different causal learning algorithms try to solve the causal discovery problem. We will start with the simplest case where there are no latent variables and then discuss how to deal with them in some situations. This categorization is associated with the causal sufficiency concept.

Causal sufficiency means that the set of observed variables contains all the causes of the system of interest, and no latent variables affect the system. This assumption is often made in causal discovery problems because it simplifies the problem of finding the causal graph that represents the system. However, this assumption rarely holds in real-world scenarios because there are often unobserved factors that influence the system in some way. [Kocaoglu, Shanmugam, and Bareinboim 2017]

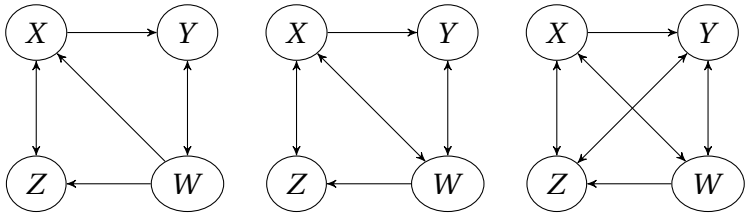
Definition 2.44 (Causal Sufficient Set) *A set of variables* $\mathbf{V}$ *is* causally sufficient *iff it includes all the variables that have a causal effect on any of the variables in* $\mathbf{V}$*. This means that no latent variables are the parents of the variables in* $\mathbf{V}$ *to influence the data generation process. A* causally sufficient *graph over* $\mathbf{V}$ *can capture all the causal relationships observed in the data. [Peter Spirtes 2010]*

In contrast, causal insufficiency refers to cases in which the set of observed variables does not contain all the causes of the system, and some latent variables exist that affect the system. This makes the causal discovery problem more challenging because the causal graph representing the system is not fully observable, and multiple causal graphs may be compatible with the data. In this case, the causal structure over the observed variables ($\mathbf{V}$) is only a sub-graph of the augmented graph that includes the unobserved variables ($\mathbf{U}$), as shown in Figure I.9, where $\mathbf{V}$ = { Smoking, Cancer } and $\mathbf{U}$ = { Genotype }.

The concept of mixed graphs (Definition 2.45) and their properties are essential for the equivalence class of constraint-based causal discovery methods and can handle causal insufficiency.R. A. Ali, T. Richardson, and Peter Spirtes 2004 We will talk about this categorization (i.e., learning algorithms for sufficient vs. insufficient causal models) in each category later in Section 3.2.1. Still, the following section discusses the graphical structures commonly used for causal insufficient algorithms.

2.4.1. Ancestral Graphs

Ancestral graphical models provide a way to effectively represent complex data-generating processes involving latent variables, which may affect the data we observe, but we cannot measure them directly. By using these models, we can include all the relevant variables in the analysis while avoiding the need to model the unobserved variables explicitly. Most of the definitions and theorems in this section are from T. Richardson and Peter Spirtes 2002 and Peter Spirtes, C. N. Glymour, and Richard Scheines 2000; otherwise, the reference is noted.



(a) Mixed Graph (not Ancestral) (b) Ancestral Graph (c) Maximal Ancestral Graph

Figure I.10.: Ancestral Graphs

Definition 2.45 (Mixed Graph) *A* mixed graph *is a graph that may contain three kinds of edges: undirected (—), directed (⟶), and bi-directed (⟷). And when it does not have any undirected edges, it is called a* directed mixed graph.

Since we have a new kind of arc (⟷), we need to define a new relationship between nodes in addition to what we defined in Section 1.1. When a bi-directed edge connects two nodes ($X \longleftrightarrow Y$), they would be called **spouses** of each other ($X \in \mathbf{Sp}(Y)$), which means X is not a parent or a child of Y. The rest of the relationships (*parent, child, ancestor*, etc.) would be the same as in Section 1.1.

Definition 2.46 (Cycle) *If* $Y \in \boldsymbol{Pa}(X)$*, and* $X \in \boldsymbol{An}(Y)$*, then a* directed cycle *is formed. And if* $Y \in \boldsymbol{Sp}(X)$*, and* $X \in \boldsymbol{An}(Y)$*, then there would be an* almost directed cycle.

As you can see in Figure I.10a, $< X, Y, W >$ is an almost directed cycle.

Definition 2.47 (Collider) *Let X, Z, and Y be three consecutive nodes on a path U. If both of these arcs are into Z ($X, Y \in \boldsymbol{Pa}(Z)$), then Z is a* collider *on U. And if the orientations are not like this, Z is a* non-collider *on U.*

For instance, in Figure I.10a, X is a collider on the path $< Z, X, W >$, but it is a non-collider on the path $< Z, X, Y >$.

To appropriately define an ancestral graph, it is necessary to first introduce the concept of **anteriors** within a graph [T. Richardson and Peter Spirtes 2002]. Consider two vertices X and Y within a mixed graph. X is described as an anterior of Y if a path C connects them such that each edge in C is undirected (Q—R) or directed towards Y ($Q \longrightarrow R$ with R between Q and Y) or $X = Y$, that is there is not any arc towards X ($Q \longleftarrow R$) or bidirectional edges ($Q \longleftrightarrow R$). The existence of such a connecting sequence classifies it as an **anterior path**, with X itself qualifying if X and Y are identical. For a set of vertices **V**, the set of all anteriors of **V** is denoted by **ant**(**V**) and is defined as:

$$\textbf{ant}(\mathbf{V}) = \{X \mid X \text{ is anterior to } Y \text{ for some } Y \in \mathbf{V}\}.$$

With the anterior concept established, we can go through the definition of the ancestral graphs. An ancestral graph $\mathcal{G}$ is a type of mixed graph that satisfies specific conditions related to anteriors:

Definition 2.48 (Ancestral Graph) *An ancestral graph $\mathcal{G}$ is a mixed graph in which the following conditions hold for all vertices X in $\mathcal{G}$:*

1. *Vertex X is not an anterior of its own parents or spouses, i.e., $X \notin \boldsymbol{ant}(\boldsymbol{Pa}(X) \cup \boldsymbol{Sp}(X))$.*
2. *If the set of neighbors of X, denoted by $\boldsymbol{Ne}(X)$, is nonempty, then there should be no directed or bidirected edges coming into X from its parents or spouses, i.e., $\boldsymbol{Pa}(X) \cup \boldsymbol{Sp}(X) = \emptyset$ when $\boldsymbol{Ne}(X) \neq \emptyset$.*

Figure I.10b shows an ancestral graph, while Figure I.10a is not an ancestral graph since $< X, Y, W >$ is an almost directed cycle.

Definition 2.49 (m-connection and m-separation) *In an ancestral graph, a path U between the nodes X, Y, is an active path or called* m-connecting *path given a set of nodes **Z**, if:*

- *every non-collider node on U is not a member of **Z***
- *every collider node on U is a member of **Z** or is an ancestor of a member of **Z***

*If there is not any m-connecting path between the two nodes X and Y given the set of nodes **Z**, they would be* m-separated *given **Z**. Hence, two disjoint sets **X** and **Y** are m-separated given **Z** if any node in **X** is m-separated from any node **Y** in given **Z**.*

For example, in Figure I.10b, the path $< X, Y, W, Z >$ is a m-connecting X and Z given the set $\{Y\}$, and it is a m-separating path given $\emptyset$, $\{W\}$, and $\{W, Y\}$ since Y is a collider, and W is a non-collider.

2.4.1.1. Maximal Ancestral Graph (MAG)

Definition 2.50 (Maximal Ancestral Graphs (MAG)) *Given an ancestral graph G, if every pair of non-adjacent nodes X and Y are m-separated given a set of nodes $\boldsymbol{Z}$, then G is a maximal ancestral graph.*

In other words, the term *maximal* refers to the fact that adding any kind of edge to the graph would change its independence set. (Equation I.6 for m-separation instead of d-separation):

Proposition 2.1 *Let $G = (\boldsymbol{V}, \mathscr{E})$ be a maximal ancestral graph and a subgraph of $G^* = (\boldsymbol{V}, \mathscr{E}^*)$. If $I_m(G) = I_m(G^*)$ then $G = G^*$*

As shown in Figure I.10 we can see Figure I.10c is a MAG, but Figure I.10b is not. The reason is that the pair (Y, Z) are the only non-adjacent nodes, and they cannot be m-separated given any set of nodes. Given the sets $\emptyset$, $\{X\}$, $\{W\}$, and $\{W, X\}$, the nodes (Y, Z) are m-connected through the paths $< Z, X, Y >$, $< Z, W, Y >$, $< Z, X, Y >$, and $< Z, X, W, Y >$ respectively. But by adding the bi-directed edge between the nodes Y and Z, the graph becomes maximal, as shown in Figure I.10c.

An inducing path is a special type of path between two nodes in a DAG that are not directly connected but still influence each other in a way that prevents them from being d-separated. This means that the path remains "active" no matter what the conditioning set is.

Definition 2.51 (Inducing Path) *Given a DAG, $G = (\boldsymbol{V}, \mathscr{E})$, a set of observed variables $\boldsymbol{U} \subseteq \boldsymbol{V}$, and two distinct variables $X, Y \in \boldsymbol{V}$, the path between X and Y is an* inducing path *relative to $\boldsymbol{U}$ if and only if:*

1. *Every non-endpoint collider on the path is in $\boldsymbol{U}$ and is an ancestor of either X or Y.*
2. *Every non-collider on the path is not in $\boldsymbol{U}$.*

The endpoints X and Y do not need to be in $\boldsymbol{U}$.

In Definition 2.51, we see no constraint on the endpoints, so we can conclude that any single edge is an inducing path relative to any set of nodes. In Figure I.10b, the path $< Z, W, Y >$ is an inducing path relative to an empty set, but it is not an inducing path relative to $\{W\}$ since W is not a collider on the path. In the same figure, we can see the path $< Z, X, W, Y >$ is an inducing path relative to $\{W, X, Y, Z\}$ since X and W are colliders on the path, and both of them are the ancestors of the endpoints.

Inducing path is important in the context of MAGs or PAGs (Section 2.4.1.2) because it shows that two nodes that are not directly connected in the original DAG can appear as if they are directly connected in the MAG or PAG. Essentially, the path induces a connection that does not exist in the original DAG. Understanding inducing paths helps to clarify that connections in MAGs and PAGs can differ from those in a DAG. In other words, just because two nodes appear adjacent in a MAG or PAG doesn't mean they are adjacent in the underlying DAG.

It is possible to present an alternative definition for MAGs regarding the definition of the inducing paths:

Theorem 2.3 *Given an ancestral graph G, if there is no inducing path between any pair of non-adjacent nodes, then G is a maximal ancestral graph.*

For instance, in Figure I.10b, it can be seen between the only pair of non-adjacent nodes (Y, Z), there is an inducing path $< Y, W, X, Z >$, so it cannot be a MAG.

2.4.1.2. Partial Ancestral Graph (PAG)

We learned in Section 1.3 that different DAGs can have the same d-separation properties and, therefore, imply the same conditional independence relations and be classified in the same I-equivalence class (Definition 1.21). This also applies to MAGs, which means that different MAGs can represent the same relations by using the m-separation criterion (Definition 2.49). This property is called *Markov equivalence*. There are different ways to describe the Markov equivalence between MAGs[Peter Spirtes and T. Richardson 1996, R. A. Ali, T. Richardson, and Peter Spirtes 2004, Zhao, Z. Zheng, and B. Liu 2005, J. Zhang and P. L. Spirtes 2012]. For our study, we only need to know that all Markov equivalent MAGs share the same skeleton and some similar edge orientations. Markov equivalence classes in MAGs can be shown with some specific partial mixed graphs (Definition 2.52), which are called Partial Ancestral Graphs (PAG) (Definition 2.53). [J. Zhang 2008]

Definition 2.52 (Partial Mixed Graphs) *A* partial mixed graph *is a graph that may contain four kinds of edges:* $\longrightarrow$, $\longleftrightarrow$, $o\!\!-\!\!o$, *and* $o\!\!\longrightarrow$.

Definition 2.53 (Partial Ancestral Graphs) *Given a MAG structure G,* $\mathcal{I}$ *is the Markov equivalence class of G. Then a partial mixed graph* P_G *is the* partial ancestral graph *if:*

- P_G *has the same skeleton as G (and any member of* $\mathcal{I}$*)*
- *an endpoint ">" is in* P_G *iff it is common in all members of* $\mathcal{I}$*.*
- *an endpoint "–" is in* P_G *iff it is common in all members of* $\mathcal{I}$*.*

Definition 2.54 (Definite Discriminating Path) *In a partial ancestral graph P_G, a path $\mathcal{T}$ is termed a* definite discriminating path *for a vertex M if and only if all the following conditions are satisfied:*

1. *$\mathcal{T}$ is an undirected path between two vertices A and B.*
2. *M is the predecessor of B on $\mathcal{T}$, and $M \neq A$.*
3. *Every vertex on $\mathcal{T}$, except for the endpoints and possibly M, is a collider on $\mathcal{T}$.*
4. *For every vertex V on $\mathcal{T}$ except for the endpoints, there is an edge $V \rightarrow B$.*
5. *A and B are not adjacent.*

This path is used to discern conditional independence relations in the graph, particularly under scenarios where traditional paths do not suffice. [Peter Spirtes 2001]

2.4.2. Causal Discovery

Definition 2.55 (Causal Discovery Problem) *Let G^* be a graphical structure (called true graph) over a set of variables* **V** *that generates a dataset D.* Causal discovery problem *consists of recovering the true graph (G^*) among all possible structures (***G** *s.t. $G^* \in$* **G***) defined over* **V**. *[Peter Spirtes, C. N. Glymour, and Richard Scheines 2000, Peter Spirtes and K. Zhang 2016]*

Many different algorithms have been proposed for solving the causal discovery problem that we will discuss later in this part. The term "solve" here means an algorithm asymptotically finds the true graph as the sample size increases.

Definition 2.56 (Soundness and Completeness of Causal Discovery Algorithms) *Suppose an algorithm can solve the causal discovery problem. In that case, it is* sound, *and if, without making extra assumptions, it can find the most informative causal graph from a given dataset, it is* complete. *[Zanga, Ozkirimli, and Stella 2022]*

Definition 2.57 (Consistency of a Causal Graph) *A causal discovery algorithm that finds a graph from a given data is* consistent *if the output converges to the true graph as the data size grows [Peter Spirtes, C. N. Glymour, and Richard Scheines 2000, Clark Glymour, K. Zhang, and Peter Spirtes 2019], i.e., the algorithm finds a graph that generates a probability distribution consistent with the input dataset. [Zanga, Ozkirimli, and Stella 2022]*

In addition to the causal sufficiency assumption (Definition 2.44), two other key assumptions are considered generally to learn a causal model: Causal Markov Assumption (Definition 2.58) and Causal Faithfulness Assumption (Definition 2.59).

Definition 2.58 (Causal Markov Assumption) *In a causal graph, the Markov assumption (Equation I.3) would be called the* Causal Markov Assumption. *[Pearl 2009] It means any variable is conditionally independent of its (direct or indirect) non-effects given its direct causes. [Koller and Friedman 2009]*

Definition 2.59 (Causal Faithfulness Assumption) *In a causal graph, the faithfulness assumption (1.25) would be called the* Causal Faithfulness Assumption. *It means that P is faithful to the causal graph G, if for every pair of X and Y that are independent given **Z**, then they are d-separated in G [Koller and Friedman 2009]*

As you may have noticed by now, there is a close relationship between Causal networks and Bayesian networks, and we explained why the former is a particular type of the latter. As a result, learning these networks involves similar principles. In Section 3, we provide a detailed discussion of both types of networks. In this section, we summarize the key points and cover additional causal discovery algorithms not discussed in Section 3.

Like Bayesian networks, causal network learning approaches can be divided into three groups: *Constraint-based* (Section 3.2.1), *Score-based* (Section 3.2.2), and *Hybrid* algorithms (Section 3.2.3).

Constraint refers to the conditional independence test to find the correct structure corresponding to the dataset. Constraint-based algorithms start with a fully connected (i.e., Complete) graph and then check the statistical dependencies between each pair of nodes to see if the arc between them is removable. Based on complying with causal sufficiency, the approaches can be divided into two groups: Causal Sufficient Algorithms (Section 3.2.1.1) like SGS [Spirtes, Glymour, and Scheines 1990] and PC [Peter Spirtes and Clark Glymour 1991] and Causal Insufficient Algorithms (Section 3.2.1.2) like Fast Causal Inference (FCI) algorithm [Peter Spirtes, Clark Glymour, Richard Scheines, et al. 1993], and Multivariate Information based Inductive Causation (MIIC) [Verny, Sella, Affeldt, et al. 2017] which can detect the presence of the latent variable. Although causal insufficient algorithms can handle latent variables, their sensitivity in choosing the significance level is considered one of their main disadvantages.

On the other hand, score-based algorithms assess the quality of different structures by assigning a score or likelihood (Section 3.2.2.1) to each structure based on how well it fits the observed data, and they select the structure that maximizes the score. Calculating these scores requires counting the occurrences of the different values variables can take, which is impossible when some variables are not observed. This is why, unlike constraint-based algorithms, they are commonly not used to learn a causal structure in the presence of latent variables. Maximal Ancestral Graphs (MAG) (Section 2.4.1.1) has been introduced precisely to fix this issue, and score-based approaches have been adapted to learn MAGs [()T. Richardson and Peter Spirtes 1999, Ogarrio, Peter Spirtes, and Joe Ramsey 2016, Triantafillou and Tsamardinos 2016]. Unfortunately, they can only cope with scoring MAGs over continuous variables. However, whenever

there is no latent variable, score-based approaches are usually considered more robust since the constraint-based algorithms are quite sensitive to errors in statistical tests.

Score-based algorithms can be classified as exact or approximate. Exact algorithms (Section 3.2.2.3) guarantee to find the graph with an optimal score even for small datasets. Some examples of exact algorithms are dynamic programming [Ott, Imoto, and Miyano 2003, Koivisto and Sood 2004] and branch-and-bound [De Campos, Zeng, and Ji 2009]. Approximate algorithms (Section 3.2.2.2) use heuristics or stochastic methods to find locally optimal graphs faster and more scalable than exact algorithms. Some examples of approximate algorithms are K2 [Cooper and Herskovits 1992], hill climbing [Remco R Bouckaert 1994], and search with tabu list [Glover 1990].

Constraint-based and score-based have some pros and cons. Thus, another group of algorithms, Hybrid methods, has been developed to take advantage of both approaches. These algorithms (Section 3.2.3) use constraint-based algorithms to reduce the size of the search space and use the score-based algorithm to find the structure that maximizes the score in the restricted space. Some examples of these algorithms are Constraint-Bayesian (CB) [M. Singh and Valtorta 1993] and Max-Min Hill Climbing (MMHC) Tsamardinos, Brown, and Aliferis 2006.

3. Learning

Throughout the previous sections, we discussed the theoretical aspects of Bayesian and Causal Networks. We showed that these networks consist of two major components: a graphical structure that displays the (causal) relationships among a group of variables and a set of parameters that quantify the strength of these relationships. Constructing these networks can be done using different methods. The first approach is to create a model manually, but this is often impractical and unreliable, particularly for large and poorly understood problems. Therefore, it is preferred to learn BNs automatically from data, using algorithms that can capture the properties of the data distribution and infer the (causal) structure.

When it comes to learning Bayesian Networks from data, there are two primary tasks: learning the parameters (Section 3.1) and learning the structure (Section 3.2).

3.1. Parameter Learning

Given some data and a fixed structure, estimating the conditional probability distributions (CPDs) is possible using parameter learning. This learning can be done using different methods, and each depends on some assumptions. We start by defining the notations, assumptions, and problem statements for BN learning. For this purpose, we follow the same procedure as Heckerman, Geiger, and David M Chickering 1995, and we focus on learning discrete variables that have a limited number of values. The goal of learning is to find the Bayesian network ($\mathscr{B}$) that matches the dataset (D) best. As mentioned in Definition 1.26 $\mathscr{B}$ consists of a BN structure $\mathscr{G} = (\mathbf{X}, \mathscr{E})$ and a set of parameters Θ. $|\mathbf{X}| = n$, which means there are n random variables in the network, therefore $\Theta = \{\theta_i\}_{i=1}^{n}$ where θ_i represents the CPT of the variable X_i (i.e. $P_{\mathscr{B}}(X_i|\mathbf{Pa}(X_i))$). This CPT can be decomposed to a set of parameters as $\theta_i = \{\theta_{ij}\}_{j=1}^{q_i}$, where θ_{ij} represents the distribution of ith variable given the jth value of the set of parents (i.e. $P_{\mathscr{B}}(X_i|\mathbf{Pa}(X_i) = \mathbf{pa}(X_i)^{(j)})$ and q_i is the joint domain size of the parents of X_i. Accordingly, θ_{ij} can be decomposed to sets of parameters as $\theta_{ij} = \{\theta_{ijk}\}_{k=1}^{r_i}$, where θ_{ijk} represent the probability of the case that X_i having its kth value and the set of its parents have its jth (i.e. $P_{\mathscr{B}}(X_i = x_i^{(k)}|\mathbf{Pa}(X_i) = \mathbf{pa}(X_i)^{(j)})$ and r_i is the domain size of the variable X_i. The data set can be denoted as $D = \{\mathbf{x}^{(m)}\}_{m=1}^{N}$ where $\mathbf{x}^{(m)}$ is the mth sample of $\mathbf{X}$, and N is the size of the data set (total number of samples).

The aim here is to find the best set of parameter values $\Theta = \{\theta_i\}_{i=1}^{n}$ for the network, given a structure $\mathscr{G}$ and dataset D. This is usually fulfilled by making Θ as close as possible to the optimal set of parameters $\hat{\Theta}$. For this purpose, a specific objective function is used to evaluate the different values that Θ can have. Different objective functions have been suggested in the literature, depending on the assumptions and the approaches used in each study, as we will introduce some of them later in this part (Sections 3.1.1 and 3.1.2).

Generally, for learning a Bayesian Network ($\mathscr{B} = (\mathscr{G}, \Theta)$) from a dataset D, following assumptions are needed [Heckerman, Geiger, and David M Chickering 1995]:

1. **IID:** The samples in D are generated *independently* from an *identical distribution.*
2. **Parameter Independence:** The parameters are *globally* and *locally* independent. Globally (Equation I.21) means the parameters for a specific variable don't influence the parameters of other variables. Locally (Equation I.22) means the parameters of a single variable are independent of each other for different values of its parents:

$$\pi(\Theta) = \prod_{i=1}^{n} \pi(\theta_i) \tag{I.21}$$

$$\pi(\theta_i) = \prod_{i=1}^{q_i} \pi(\theta_{ij}) \tag{I.22}$$

   where π is a *prior* distribution.
3. **Parameter Modularity:** If a variable (X_i) in two different graphs has the same set of parents($\mathbf{Pa}(X_i)$) then θ_i is equal in both BNs ($\theta_i = P_{\mathscr{B}}(X_i|\mathbf{Pa}(X_i))$).
4. **Dirichlet Prior:** The prior of a parameter set θ follows a Dirichlet distribution with the hyperparameters $\{\beta_k\}_{k=1}^{r_i}$:

$$Dirichlet(\theta_{ij}|\beta_1, \cdots, \beta_{r_i}) = \frac{\Gamma(\sum_{k=1}^{r_i} \beta_k)}{\prod_{k=1}^{r_i} \Gamma(\beta_k)} \prod_{k=1}^{r_i} \theta_{ijk}^{\beta_k - 1} \tag{I.23}$$

5. **Complete Data**: The data set is complete, which means there is no missing value.

To begin, we will explain the fundamental concepts underlying maximum likelihood estimation.

3.1.1. Maximum Likelihood Estimation (MLE)

We will present the definitions of this section by an example, as done in Koller and Friedman 2009 did. Consider an experiment of tossing a biased coin several times, creating a dataset, and adding a sample for the observation each time we flip it. We intend to estimate the probability of observing the head (H) or the tail (T) of the next toss based on the dataset. To formalize this process, it is considered that there is a same distribution where the results of the tosses are sampled from. $x^{(1)}, .., x^{(m)}$ represent the outcomes observed from the first flip to the mth flip (sample), and they are independent and identically distributed (iid). The probability for observing H is θ, and for T is $(1-\theta)$. The goal is to find a suitable value for θ. The set of all possible

parameters (θ) can be defined as Θ, where $\theta \in [0,1]$. Let's say our current samples are H,T,T,H,T,T,T,H,T,T. The probability of observing this sequence is:

$$P(<H,T,T,H,T,T,T,H,T,T>,\theta) =$$

$$\theta(1-\theta)(1-\theta)\theta(1-\theta)(1-\theta)(1-\theta)\theta(1-\theta)(1-\theta) = \theta^3(1-\theta)^7.$$

This equation shows as we change θ, the probability of observing this sequence in the dataset changes. This function can also be called a likelihood function since it presents the likelihood of the data given the parameter $L(D|\Theta)$. It is useful to find out how changing the parameter will affect the likelihood of the dataset, then we can select the value that maximizes the likelihood, which is called the *maximum likelihood estimator*. This value is defined as:

$$\hat{\theta}^{\text{MLE}} = \arg\max_{\theta\in\Theta} L(D \mid \theta) \tag{I.24}$$

We can apply the MLE technique to estimate the parameters in a Bayesian Network. The idea is the same as for a single variable: we want to find the parameter values under which the observed dataset is most probable. The difference is that we have to do this for every node X_i and its CPT, given its parents in $\mathcal{G}$. So, we can rewrite the equation I.24 as:

$$\hat{\Theta}^{MLE} = \operatorname*{Argmax}_{\Theta} L(D|\mathcal{G},\Theta) \tag{I.25}$$

Where $\hat{\Theta} = \{\hat{\theta}_1 \cdots \hat{\theta}_n\}$, and $L(D|\mathcal{G},\Theta)$ can be defined as:

$$\begin{aligned} L(D|\mathcal{G},\Theta) &= \prod_{m=1}^{N} P(x^{(m)}|\mathcal{G},\Theta) \\ &= \prod_{m=1}^{N}\prod_{i=1}^{n} P(X_i = x_i^{(m)}|\mathbf{Pa}(X_i) = \mathbf{pa}(X_i)^{(m)},\Theta) \\ &= \prod_{m=1}^{N}\prod_{i=1}^{n} \theta_{ij_mk_m} \end{aligned} \tag{I.26}$$

Here $x^{(m)}$ represents the data observed in the mth sample, and $\mathcal{G}$ represents the structure of the Bayesian Network. The overall likelihood is the product of the probabilities of the individual observations, each calculated according to the network structure and the parameters Θ. $\mathbf{pa}(X_i)^{(m)}$ refers to the observed values of the parents of X_i in the mth sample of D. The likelihood can also be written in terms of the parameters θ_{ijk} associated with each variable X_i and its parent configurations as:

$$L(D|\mathcal{G},\Theta) = \prod_{i=1}^{n}\prod_{j=1}^{q_i}\prod_{k=1}^{r_i} (\theta_{ijk})^{N_{ijk}} \tag{I.27}$$

where N_{ijk} is the number of samples in which X_i takes the kth value and $\mathbf{Pa}(X_i)$ is in the jth configuration.

By calculation the logarithm of Equation I.27 we would have the *log-likelihood*:

$$LL(D|\mathcal{G},\Theta) = \sum_{i=1}^{n}\sum_{j=1}^{q_i}\sum_{k=1}^{r_i} N_{ijk} log\theta_{ijk} \tag{I.28}$$

By the assumption of (global) parameter independence, the Equation I.25 can be solved for each parameter separately:

$$\hat{\theta}_i = \underset{\theta_i}{\text{Argmax}} \sum_{j=1}^{q_i}\sum_{k=1}^{r_i} N_{ijk} log\theta_{ijk},\ 1 \le i \le n \tag{I.29}$$

Since $\theta_i = \{\theta_{ij}\}_{j=1}^{q_i} = \{\theta_{i1},\cdots,\theta_{iq_i}\}$, then, in the same way as mentioned above, we can rewrite Equation I.29 as follows:

$$\hat{\theta}_{ij} = \underset{\theta_{ij}}{\text{Argmax}} \sum_{k=1}^{r_i} N_{ijk} log\theta_{ijk},\ 1 \le i \le n,\ 1 \le j \le q_i \tag{I.30}$$

Also, we know that $\theta_{ij} = \{\theta_{ijk}\}_{k=1}^{r_i} = \{\theta_{ij1},\cdots,\theta_{ijr_i}\}$, where $\theta_{ijk} \ge 0$, and $\sum_{k=1}^{r_i}\theta_{ijk} = 1$. This equation can be transformed as $\theta_{ijr_i} = 1 - \sum_{k=1}^{r_i-1}\theta_{ijk}$ and if we substitute it in I.30 we would have:

$$\hat{\theta}_{ij} = \underset{\theta_{ij}}{\text{Argmax}}[\underbrace{(\sum_{k=1}^{r_i-1} N_{ijk} log\theta_{ijk}) + N_{ijr_i} log(1 - \sum_{k=1}^{r_i-1}\theta_{ijk})}_{LL_{ij}(D|\theta_{ij})}] \tag{I.31}$$

To find the value of $\hat{\theta}_{ij}$ in this equation, we need to find the partial differential of the term inside the bracket and set it equal to zero:

$$
\begin{aligned}
&\frac{\partial LL_{ij}(D|\theta_{ij})}{\partial \theta_{ijk}} = 0 \Longrightarrow \\
&\frac{\partial LL_{ij}(D|\theta_{ij})}{\partial \theta_{ijk}} = \frac{N_{ijk}}{\theta_{ijk}} - \frac{N_{ijr_i}}{(1-\sum_{k'=1}^{r_i-1}\theta_{ijk'})} = \frac{N_{ijk}}{\theta_{ijk}} - \frac{N_{ijr_i}}{\theta_{ijr_i}} = 0 \Longrightarrow \\
&\frac{N_{ijk}}{\theta_{ijk}} = \frac{N_{ijr_i}}{\theta_{ijr_i}}, \ \forall k \in \{1,\cdots,r_i-1\} \Longrightarrow \\
&\frac{N_{ijk}}{\theta_{ijk}} = \frac{N_{ijr_i}}{\theta_{ijr_i}} = \frac{\sum_{k=1}^{r_i} N_{ijk}}{\sum_{k=1}^{r_i}\theta_{ijk}} = \sum_{k=1}^{r_i} N_{ijk} \Longrightarrow \\
&\hat{\theta}_{ijk}^{MLE} = \frac{N_{ijk}}{\sum_{k'=1}^{r_i} N_{ijk'}}
\end{aligned}
\tag{I.32}
$$

Equation I.32 means that we can find the parameters that maximize the likelihood by counting the frequency of each combination of each X_i and $\mathbf{Pa}(X_i)$ in dataset D. However, this method requires a large dataset to be accurate; otherwise, the estimation might be rough.

3.1.2. Bayesian Estimation

The Bayesian approach considers both the data D and the parameters Θ as random variables with a joint probability distribution $P(D,\Theta|\mathscr{G})$. This implies that there is a prior distribution $\pi(\Theta|\mathscr{G})$ over the possible values of Θ, which represents our initial belief before observing the data. This belief is updated with the data using the Bayes rule to obtain a posterior distribution $P(\Theta|\mathscr{G},D)$.

$$
P(\Theta|\mathscr{G},D) = \frac{P(D|\mathscr{G},\Theta)\pi(\Theta|\mathscr{G})}{P(D|\mathscr{G})} \propto P(D|\mathscr{G},\Theta)\pi(\Theta|\mathscr{G}) \tag{I.33}
$$

As we can see, unlike the MLE approach which can yield parameter estimates of zero in the case of unobserved events, the Bayesian approach allows for the incorporation of prior knowledge to provide more robust estimates for parameters, even when some events are not observed in the data. In Equation I.33 denominator of the fraction is a constant value since it is not a function of Θ and it is not needed for finding the Θ maximizing the posterior probability $\Theta|\mathscr{G},D$, which is why it is removable. So based on this equation, we can see that the posterior probability proportion to the likelihood ($P(D|\mathscr{G},\Theta)$) and the prior distribution($\pi(\Theta|\mathscr{G})$). From Equation I.27, we have the likelihood part for the multinomial distribution:

$$
P(D|\mathscr{G},\Theta) = \prod_{i=1}^{n}\prod_{j=1}^{q_i}\prod_{k=1}^{r_i}(\theta_{ijk})^{N_{ijk}} \tag{I.34}
$$

Based on Equations I.33 and I.34, since the prior with the Dirichlet distribution is a conjugate prior for the likelihood with a multinomial distribution, it is possible to select this distribution for the prior (Equation I.35) and posterior (Equation I.36) distributions.

$$\pi(\Theta|\mathcal{G}) \propto \prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} (\theta_{ijk})^{\alpha_{ijk}} \tag{I.35}$$

Then from Equation I.33 we would have:

$$P(\Theta|\mathcal{G}, D) \propto P(D|\mathcal{G}, \Theta)\pi(\Theta|\mathcal{G}) \propto \prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} (\theta_{ijk})^{N_{ijk}+\alpha_{ijk}} \tag{I.36}$$

Where the hyperparameters $\alpha_{ijk} > 0$.

The same way as finding the MLE, we can compute the θ^{MAP} (maximum a posteriori) by calculating the differentiation of the log of Equation I.36:

$$\hat{\theta}_{ijk}^{MAP} = \frac{N_{ijk} + \alpha_{ijk}}{\sum_{k'=1}^{r_i} N_{ijk'} + \alpha_{ijk'}} \tag{I.37}$$

3.2. Structure Learning

Learning the parameters is relatively easy given a fixed structure and can be done using a closed-form formula under certain assumptions. However, learning the structure is more challenging, as it is an NP-hard problem [David Maxwell Chickering 1996] that requires searching over a large space of possible graphs. To solve this problem, various algorithms have been proposed, which can be categorized into three classes: Constraint-based (Section 3.2.1), Score-based (Section 3.2.2), and Hybrid (Section 3.2.3).

Constraint-based algorithms locate the skeleton and v-structures of a graph by using conditional independence tests based on rules derived from the properties of DAGs. On the other hand, score-based algorithms use a scoring function to evaluate the fit of different graphs to the data and search for the optimal graph using heuristic methods. Hybrid algorithms combine both approaches, using conditional independence tests to reduce the search space of graphs and then using a scoring function to find the best graph within that space. These algorithms are designed to improve the accuracy and efficiency of learning.

The performance of these algorithms depends on various factors, such as the amount and quality of data, the performance metric, and the goal of learning. For instance, learning a complete Bayesian network may be necessary if the aim is to perform probabilistic inference or causal discovery. However, learning a subset of the

network related to a target variable may be sufficient if the goal is classification or prediction. Additionally, these algorithms' capabilities can be enhanced by adding features such as confidence assessment, latent variable detection, and consistency checking.

3.2.1. Constraint-based Approaches

Constraint-based algorithms for structure learning primarily rely on conditional independence tests to discover the relationships between variables and to construct a graph that captures these relations. Some approaches within this category may also employ information-theoretic tools like mutual information instead, or in addition to, traditional statistical tests. Constraint-based algorithms usually assume that the graph shows causal relationships under some assumptions, such as *causal sufficiency* (Definition 2.44 and faithfulness (Definition 1.25). The learned graph reflects the conditional dependences and independences of the underlying probability distribution, which means when two nodes are not connected to each other, there is a conditional independence between them. The direction of the edges is decided by using some rules (e.g., Meek's rules[Meek 1995]) that make the graph agree with the independence tests. Constraint-based algorithms do not give a single DAG but a set of DAGs that are I-equivalent (Definition 1.21). As said before, these sets are the equivalence classes and are usually represented by a complete partially directed acyclic graph (CPDAG) or partial ancestral graph (PAG) (Section 2.4.1.2). Constraint-based algorithms differ in the type of statistical test, the methods for directing the graph, and the principles they use to improve the reliability of their tests. The statistical test also relies on a significance level α, which influences the number of edges in the learned graph.

There are two groups of constraint-based approaches in terms of causal sufficiency (Definition 2.44). The causal sufficient ones, like SGS [Spirtes, Glymour, and Scheines 1990], PC [Peter Spirtes and Clark Glymour 1991], and their variants. In addition to them, we will also represent the causal insufficient structure learning algorithms, such as FCI [Peter Spirtes, C. N. Glymour, and Richard Scheines 2000] and its variants. Table I.2 summarized all the algorithms we cover in this section sorted by the year. In this table, the outputs are abbreviated: CPDAG (Definition 2.21), PAG (Definition 2.53), and PC (Parents and Children), which is defined during the presentation of the algorithms. The sufficient column shows whether the algorithm is causally sufficient or insufficient.

Algorithm	Name	Year	Sufficient	Output
SGS	Spirtes-Glymour-Scheines	1990	Yes	CPDAG
PC	Peter and Clark	1991	Yes	CPDAG
FCI	Fast Causal Inference	1995	No	PAG
HITON-PC	Hiton Parents & Children	2003	Yes	PC
MMPC	Max–Min Parents Children	2003	Yes	PC
CPC	Conservative PC	2006	Yes	CPDAG
SI-HITON-PC	Semi-interleaved Hiton Parents & Children	2010	Yes	PC
RFCI	Really Fast Causal Inference	2012	No	PAG
CFCI	Conservative Fast Causal Inference	2012	No	PAG
FCI+	Fast Causal Inference+	2013	No	PAG
PC-Stable	Peter and Clark—Stable	2014	Yes	CPDAG
FCI-Stable	Fast Causal Inference Stable	2014	No	PAG
VCSGS	Very Conservative SGS	2014	Yes	*NA*
PC-Max	Peter and Clark—MAX	2016	Yes	CPDAG
MIIC	Multivariate Information-based Inductive Causation	2017	No	CPDAG

Table I.2.: Constraint-based Structure Learning Algorithms

3.2.1.1. Causal Sufficient Algorithms

One of the first constraint-based algorithms was **SGS** [Spirtes, Glymour, and Scheines 1990], which has some principled rules that created many different algorithms in the same field. This algorithm begins from a fully connected graph and tries to find the optimal structure in three steps:

1. For every pair of nodes, it checks if, based on the dataset, they are adjacent or not. This step, called *adjacency search*, checks if two nodes are independent given any subset of the other nodes to remove the edge between them. The output of the first step would be the *skeleton*:

   if $\exists \mathbf{S} \subseteq \mathbf{V} \backslash \{X, Y\} s.t. X \perp\!\!\!\perp Y | \mathbf{S}$; $\Longrightarrow$ The edge between X—Y must be removed.

2. This step is called *v-structure search*, which means it looks for triples like X—Y—Z to see if it is possible to direct them as $X \longrightarrow Y \longleftarrow Z$. A *PDAG* would be the output of this step:

   if X—Y and Y—Z are connected, but X and Z are not connected, and $X \not\perp\!\!\!\perp Z | \mathbf{S} \bigcup Y$; for any $\mathbf{S} \subseteq \mathbf{V} \backslash \{X, Y, Z\} \Longrightarrow$ direct the triple as $X \longrightarrow Y \longleftarrow Z$

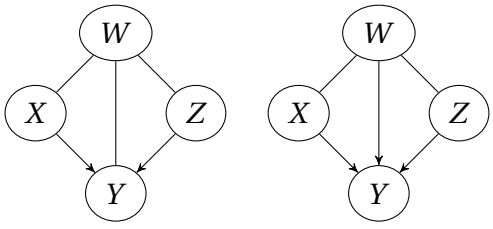
3. The final part would be the *orientation step*, which directs the remaining undirected edges. The algorithm checks if orienting an edge would create a *cycle* or a *new v-structure*. If that is not the case, the edge would be oriented; otherwise, it would be a forbidden orientation. The edges would be checked repeatedly until

no more orientation would be possible for any of them. The final output would be a *CPDAG*.

As we see, the SGS algorithm is not time efficient since it tests the independence for all edges and for too many conditioning sets (even every possible subset), which makes it an impractical solution. To solve this problem, Peter Spirtes and Clark Glymour 1991 proposed the **PC algorithm**, which can be considered an enhanced version of the SGS algorithm. PC reduces the number of tests by using their results in further tests. The same as SGS, it starts with a complete graph and performs the adjacency search step by increasing the size of the conditioning set either until two nodes are independent or until all the nodes (except the edge's two endpoints) are considered in the conditioning set. Whenever X&Y are conditionally independent, their separating set will be stored as $Sepset_{XY}$. Another difference with SGS is that the adjacency step would only search the neighbors of two nodes as the possible members of the Sepset, which will reduce the number of tests significantly in comparison with SGS. Another improvement is in the v-structure search step; it uses the Sepsets found during the first step. For any triple X—Y—Z, if Y is not in the Sepset of X and Z, it is directed $X \longrightarrow Y \longleftarrow Z$. For the orientation step, it uses the Meek's rules [Meek 1995]:

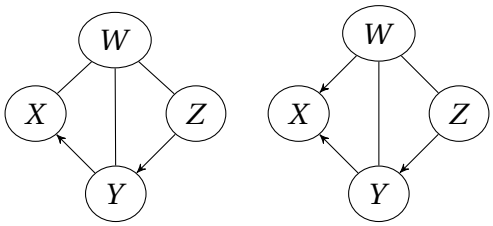
- **Rule 1:** (*forbidden v-structures*)If there is a triple like $X \longrightarrow Y$—Z (X and Z are not adjacent), then orient $Y \longrightarrow Z$ since the reverse direction will create a new v-structure (they had not been created with Rule 1) and it is forbidden.
- **Rule 2**: (*avoid cycle creation*) If an undirected edge connects X and Y and there is a directed path between them ($X \longrightarrow \cdots \longrightarrow Y$), then orient the edge between them from X to Y ($X \longrightarrow Y$) since the opposite direction would add a cycle.
- **Rule 3**: (*avoid cycle & new v-structure creation*) If there is a structure like Figure I.11a, orient as Figure I.11b. Since if we have the opposite direction, to avoid creating cycles, we need to apply Rule 2 twice, and then we would be obliged to have a v-structure.
- **Rule 4**: (*avoid cycle & new v-structure creation*) If there is a structure like Figure I.12a, orient as Figure I.12b. And the reason is exactly as Rule 3.

These four rules are applied iteratively until there is no more edge remaining to be oriented.



(a) Before (b) After

Figure I.11.: Meek's 3rd rule



(a) Before (b) After

Figure I.12.: Meek's 4rd rule

The PC algorithm is more efficient than the SGS algorithm because it avoids testing unnecessary separators that have already been shown to be independent. Still, it is less stable since, as we mentioned before, the results of the tests would be used again later in the algorithms so that any mistake in removing an edge can lead to another error later in the other steps. These problems resulted in the definition of other variants of PC.

Joseph Ramsey, J. Zhang, and Peter Spirtes 2006 weakened the faithfulness assumption in PC and SGS and would discriminate between faithful and unfaithful v-structure using their new assumptions. Hence, they called the algorithm a **Conservative PC (CPC)** since it is more cautious in the orientation of the edges than the PC algorithm and reduces the errors in finding the correct edges. J. Zhang and Peter Spirtes 2008 expressed that it is possible to apply this weakened faithfulness in the adjacency step as well as the v-structures step, and Peter Spirtes and J. Zhang 2014 used this assumption to propose their algorithm called **Very Conservative SGS (VCSGS)**. Colombo, Maathuis, et al. 2014 showed that one important factor in PC mistakes is related to the order in which conditional independence tests are done. They demonstrated that all three steps of the PC algorithms (and other algorithms with similar steps) are sensitive to the mentioned ordering. As a result, they modified the steps of PC to consider the order of the independence tests and proposed their algorithm called **PC-Stable**. As we said, PC selects the Sepsets from the adjacency step for performing the v-structure step. Joseph Ramsey 2016 proposed an algorithm called **PC-MAX**, which considers the p-value of the Sepsets and selects the one with the highest value. In addition, this approach helped them to decide how to orient in the case of conflicts of v-structures, as shown in Figure I.13. In these situations, they orient the conflicting edge in the direction of a v-structure, which has a higher p-value. They also increased the efficiency by parallelizing the first and second steps of the PC and considered the order of independence tests as well as PC-Stable.

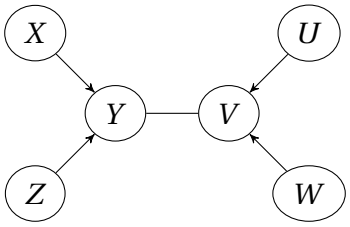


Figure I.13.: V-structures Conflicts

The constraint-based algorithms mentioned so far are called global discovery algorithms because they try to learn the whole graph structure at once. On the other hand, there is another group called local discovery algorithms, which aims to learn the local structure of each variable separately. Some of these algorithms consider the *Markov Blanket* of each node (denoted as $\boldsymbol{MB}(X)$) as the learned local structures [Margaritis and Thrun 1999, Margaritis et al. 2003, Yaramakala and Margaritis 2005]. $\boldsymbol{MB}(X)$ is a minimal conditioning set that makes X independent of all other nodes (except $\boldsymbol{MB}(X)$) [Pearl 1988]. There is another group of local discovery algorithms that take the set of parents and children of each node in the structure (denoted as $\boldsymbol{PC}(X)$) as the learned local structures e.g., Min-Max Parents Children (**MMPC**) by Tsamardinos, Aliferis, and A. Statnikov 2003, **HITON-PC** by Aliferis, Tsamardinos, and A. Statnikov 2003, Aliferis, Tsamardinos, A. R. Statnikov, et al. 2003, and **SI-HITON-PC** by Aliferis, A. Statnikov, Tsamardinos, et al. 2010. A comprehensive framework is established in the last mentioned article, which demonstrates how the three mentioned algorithms learn the $\boldsymbol{PC}(X)$, which includes three different strategies: *inclusion*, *elimination*, and *interleaving* the other two strategies. These strategies modify the $\boldsymbol{PC}(X)$ (add/remove member) by assessing the association between X and other nodes in the structure.

3.2.1.2. Causal Insufficient Algorithms

The constraint-based algorithms we described until now are all causally sufficient, meaning they assumed there is no latent variable in the data, which is not true in most real problems. So, we need also to discuss the algorithms in which this assumption is not applied. Since these variables are hidden, or there is not enough information, it is harder to learn the structure with their presence. Based on what we explained in Section 2, it is not possible for DAGs to represent the information about the presence of the latent variable. That is why the ancestral graphs (Section 2.4.1) are used in this field, which has a bidirected arc between two nodes that are dependent because of a latent confounder.

One of the causal insufficient algorithms is **Fast Causal Inference (FCI)** [Peter Spirtes, Clark Glymour, Richard Scheines, et al. 1993, Peter Spirtes, C. N. Glymour, and Richard Scheines 2000]. They considered some assumptions, including causal Markov and causal faithfulness. They proposed different definitions, including the

Algorithm 1: Fast Causal Inference (FCI)

Input: a dataset including the observable variables (**V**)
Output: a POIPG

$G \leftarrow$ Complete undirected graph on **V**
$i \leftarrow 0$
while *There is a connected ordered pair* (X, Y), *and* $|\{\mathbf{Ne}_G(X) \setminus Y)\}| \geq i$ **do**
  for *all connected ordered pair* (X, Y) *with* $|\{\mathbf{Ne}_G(X) \setminus Y)\}| \geq i$ **do**
    $\mathbf{S} \leftarrow \{S | S \subseteq \mathbf{Ne}_G(X) \setminus Y)\}$
    for *each* $S \in \mathbf{S}$ *s.t.* $|S| = i$ **do**
      if $d\text{-}sep_G(X; Y \mid S)\}$ **then**
        remove edge between X and Y from G
        add S to **Sepset**(X, Y)
  $i \leftarrow i + 1$
for *each triple* X—Z—Y **do**
  if $Y \notin \mathbf{Ne}_G(X)$, *and* $Z \notin \mathbf{Sepset}(X, Y)$ **then**
    orient X—Z—Y as $X \longrightarrow Z \longleftarrow Y$
for *each connected pair* (X, Y) **do**
  $\mathbf{S} \leftarrow \{S | S \subseteq$ **Possible D-Sep**$(X, Y) \setminus (X, Y)\}$
  if $\exists S$ *s.t.* $d\text{-}sep_G(X; Y \mid S)\}$ **then**
    remove edge between X and Y from G
    add S to **Sepset**(X, Y)
Orient each edge as o—o
for *each* $X * — * Z * — * Y$ **do**
  if $Y \notin \mathbf{Ne}_G(X)$, *and* $Z \notin \mathbf{Sepset}(X, Y)$ **then**
    orient $X * — * Z * — * Y$ as $X * \longrightarrow Z \longleftarrow * Y$
while *it is not possible to orient any more edge* **do**
  if $\exists X * — * Y$ *and,* $\exists X \longrightarrow \cdots \longrightarrow Y$ **then** orient $X * — * Y$ as $X * \longrightarrow Y$
  if $\exists M * \longrightarrow \underline{N *} — * P$ *i.e., The graph does not contain* $M * \longrightarrow N \longleftarrow * P$ *(*M *and* P *does not collide on* N*)* **then** orient it as $M * \longrightarrow N \longrightarrow P$
  if V *is a collider in* $< U, V, W >$, *and* $U \notin \mathbf{Ne}_G(W)$, *and* $V \in \mathbf{Ne}_G(T)$, *and* T *is not a collider in* $< U, T, W >$ **then** orient $V * — * T$ as $V \longleftarrow * T$
  if $\exists A \longleftarrow * B, A \longrightarrow C$, *and* $Co — * B$ **then** orient $Co — * B$ as $C \longleftarrow * B$
  if D *is a definite discriminating path (Definition 2.54) between* J *and* K *for* H, F *is neighbor of* K *on* D, *and* $< F, H, K >$ *is a triangle* **then**
    if $H \in \mathbf{Sepset}(J, K)$ **then**
      H is not a collider (orient as $F * — * \underline{H *} — * K$)
    else
      orient $F * — * H * — * K$ as $F * \longrightarrow H \longleftarrow * K$
return G

Inducing Paths, *Inducing Path Graphs* (*IPG*), and *Partially Oriented Inducing Path Graphs* (*POIPG*). IPG has directed and bidirected edges, while POIPG has two more kinds of edges o—o and $o \longrightarrow$ means the algorithm is not sure about the direction of the edge. They also used the $*$ as an endpoint to consider all three possibilities for an endpoint $(o, -, >)$. Based on these definitions, they introduced new d-separating sets called *D-Sep* for IPG and *Possible D-Sep* for POIPG. In general, FCI steps are the same as PC, while PC is based on DAGs, and FCI performs the steps on POIPGs. For example, in the adjacency step, PC finds the conditioning set (Sepset) as a subset of the parents of the nodes. In FCI, the Sepset is a subset of the D-Sep, which is not necessarily the neighbors of the nodes, and it is possible that they will not be adjacent in the graph. FCI begins from a complete undirected graph (like PC) and performs independence tests on each ordered pair of nodes conditioning on their neighbors to find the initial skeleton. Then, it changes the edges of the structure from the adjacency step to o—o. Afterward, for each triple $X * — * Y * — * Z$ that has the conditions of being a v-structure, it converts the triple to $X* \longrightarrow Y \longleftarrow *Z$. By considering the orientations in the output structure, they find some nodes are not definitely in the D-Sep; hence, the rest of the nodes are members of a set called Possible D-Sep. In the next step, FCI redoes the adjacency step by using subsets of Possible-D-Sep as the conditioning sets. Then, again, it checks the skeleton to find the v-structures. In the end, it repeats its orientation step, which is more complicated than PC since, in the final results, there are more kinds of edges between the nodes. Algorithm 1 shows the steps of FCI in detail. More information is available in Peter Spirtes 2001.

The modifications made to the adjacency step of the FCI algorithm made it mainly slower than the PC algorithm. Colombo, Maathuis, Kalisch, et al. 2012 tried to solve this problem by proposing the **Really Fast Causal Inference (RFCI)**, which performs the same adjacency step as PC (considering only the parents of the two nodes in the conditioning sets). Also, instead of two adjacency steps in FCI, they reduced it to one step. And they made some modifications in v-structures and orientation steps. They showed that the speed of their algorithm is much higher than the normal FCI, while they usually have the same structures as FCI. Colombo, Maathuis, Kalisch, et al. 2012 tried to weaken the faithfulness assumption (Like what Joseph Ramsey, J. Zhang, and Peter Spirtes 2006 did to produce Conservative PC) in the v-structures step and defined the **Conservative FCI (CFCI)** algorithm. As a result, they distinguished the v-structures by being ambiguous or not and accepted the non-ambiguous ones. Compared with FCI, the separating sets are possibly smaller in CFCI, and it has fewer arcs and more edges in the final structure. But in comparison with RFCI, it has more similar orientations with FCI. Some other algorithms tried to make the FCI faster. Claassen, J. Mooij, and Heskes 2013 proposed the **FCI+**, which increased the speed of the adjacency step by focusing on the Sepsets with ancestors instead of just the parents of the nodes. Colombo, Maathuis, et al. 2014 did the same modifications as PC-Stable to propose the **FCI-Stable** by making the result independent of the ordering considered in FCI.

Verny, Sella, Affeldt, et al. 2017 proposed Multivariate Information based Inductive Causation (**MIIC**), a causal insufficient method that integrates constraint-based and information-theoretic approaches to uncover causal structures from observational data. This method is capable of learning a wide range of graphical models, including both causal and non-causal relationships while considering the effects of latent variables. It begins with a complete graph and progressively refines this by removing edges that do not show significant indirect informational contributions. To determine the confidence of each edge, **MIIC** employs a permutation-based approach, where the original dataset is randomly permuted to create a distribution for edge-specific confidence assessments. This helps ensure that the connections retained in the final graph are statistically significant and not artifacts of random data variations. Following this, the remaining graph structure is refined by orienting v-structures based on the established patterns of causality observed in the data.

3.2.2. Score-based Approaches

The score-based algorithms, which this thesis will mainly concentrate on, are another group of structure learning methods that use a scoring function to evaluate and compare different graphs given the data. The scoring functions measure how well a graph fits the data and capture the trade-off between complexity and accuracy. Score-based algorithms search for the graph with the highest score among the huge number of possible graphs. The search strategy determines how the algorithm explores the space of graphs and how it avoids getting stuck in a local optimum. The search space can be either the space of DAGs, the equivalence classes (Definition 1.21), or the node-ordering (Definition 1.5). Score-based algorithms can also be categorized by their accuracy, which is exact or approximate. Exact algorithms guarantee to find the optimal graph, while approximate algorithms use heuristics to find a good but not necessarily optimal graph. They studied the independence between selecting a score and an algorithm, which is why score-based algorithms in many BN libraries are flexible and can use different scoring functions. In this section, we first discuss different scoring functions; then, we go through some of the most famous score-based structure learning algorithms. Table I.3 represents all the algorithms we discuss in this section sorted by their year. It also shows if the method is an exact (Section 3.2.2.3) or approximate (3.2.2.2) algorithm. It expresses each method's score function besides the output structure they return: DAG, CPDAG (Definition 2.21), and MAG (Definition 2.50).

Algorithm	Name	Year	Exact	Score	Output
K2	K2	1992	No	K2	DAG
HC	Hill Climbing	1994	No	BDe	DAG
Tabu	Hill Climbing with Tabu List	1995	No	K2 & BIC	DAG
GES	Greedy Equivalence Search	2002	No	BDeu	CPDAG
OptOrd	Dynamic Programming	2005	Yes	BDeu	DAG
BnB	Breach and Bound	2009	Yes	AIC & BIC	DAG
CPBayes	Constraint Programming	2015	Yes	BDeu & BIC	DAG
GSMAG	Greedy Search for MAGs	2016	No	BIC for MAG	MAG
FGES	Fast Greedy Equivalence Search	2017	No	BDeu & BIC	CPDAG
MAHC	Model Averaging Hill Climbing	2022	No	BIC	DAG

Table I.3.: Score-based Structure Learning Algorithms

3.2.2.1. Scoring Functions

The search algorithm's cost depends on the problem's properties. The main factor is how many neighboring networks we need to evaluate at each step. To score each of these network structures, we need to go through all the data and calculate sufficient statistics for the new structure. This can be very costly and is the main expense in any structure learning algorithm. This is where the **score decomposability** property becomes helpful, which represents that the scores can be split into a sum of terms, one for each variable X_i.

Definition 3.1 (Score Decomposition) *A* decomposable *scoring function is a function that can be written as the sum of scores for each node in the network, where the score of a node only depends on the node itself and its parents:*

$$Score(\mathcal{G}, D) = \sum_{i=1}^{n} Score(X_i | Pa(X_i), D) \tag{I.38}$$

Definition 3.1 expresses that the local score ($Score(X_i|Pa(X_i), D)$) depends on the dataset, which means a local change that adds, deletes, or reverses an edge does not affect most of the network's score. So, it is needed to find the local changes in the score instead of computing the score of the whole structure again. The score component stays the same for families that do not change their composition.

Definition 3.2 (Score Equivalence) *Let* $Score(\mathcal{G}, D)$ *be a specific scoring function, and* $\mathcal{G}$ *and* $\mathcal{G}'$ *be two I-equivalent networks, then the function satisfies* score equivalence *if* $\forall D$, $Score(\mathcal{G}, D) = Score(\mathcal{G}', D)$.

Definition 3.3 (Score Consistency) *Let $\mathcal{G}^*$ be a perfect map for the distribution P^* that generates the data D. A scoring function is* consistent *if, as the data size grows indefinitely ($N \longrightarrow \infty$) with probability tending to 1 (over the possible data sets D), then **i)** The score is maximized by the structure $\mathcal{G}^*$, **ii)** Any structure $\mathcal{G}$ that is not I-equivalent to $\mathcal{G}^*$ has a lower score than $\mathcal{G}^*$. [Koller and Friedman 2009]*

As you can imagine, one of the most crucial choices we have to make in this field is the type of score function. Two main groups of scores can be used to classify data: Bayesian scores and information-theoretic scores. Bayesian approach is built on the concepts discussed in the previous section regarding parameter estimation. As mentioned earlier, the Bayesian method uses a distribution to represent uncertainty. Since we are uncertain about both the structure and parameters, we need to define a prior probability distribution for the structure ($P(\mathcal{G})$) that assigns probabilities to different graph structures and a prior probability distribution for the parameter ($P(\theta|\mathcal{G})$) that assigns probabilities to different parameter values, given the graph. Then, the Bayesian scoring functions produce a posterior probability for the structure by considering the prior knowledge. And unlike the constraint-based methods, they can assign a score to each possible structure using those posterior probability distributions.

In this section, we will highlight some of the most used Bayesian scores in literature, including Bayesian Dirichlet (BD) and K2 score [Cooper and Herskovits 1992], Bayesian Dirichlet Equivalence (BDe) [Heckerman, Geiger, and David M Chickering 1995], and Bayesian Dirichlet Equivalence Uniform (BDeu) [Buntine 1991]. Afterward, we also discuss information-theoretic scoring functions and introduce some of the most common among them, such as the likelihood score, Akaike information criterion (AIC), and Bayesian information criterion (BIC).

Definition 3.4 (Bayesian Score) *Let $P(\mathcal{G})$ and $P(\theta|\mathcal{G})$ be the prior probability distributions for the structure and the parameter, respectively, and consider $P(D|\mathcal{G})$ as the likelihood of the data given the bayesian network $B = (\mathcal{G}, \Theta)$. We can start calculating the* Bayesian Score *with the Bayes rule:*

$$P(\mathcal{G}|D) = \frac{P(D|\mathcal{G})P(\mathcal{G})}{P(D)} \tag{I.39}$$

The denominator is a normalizing factor that cannot differentiate between structures. So the Bayesian Score *can be defined as follows:*

$$Score_B(\mathcal{G}, D) = logP(D|\mathcal{G}) + logP(\mathcal{G}) \tag{I.40}$$

The term $P(D|\mathcal{G})$ accounts for our uncertainty regarding the parameters:

$$P(D|\mathcal{G}) = \int_\Theta P(D|\theta, \mathcal{G})\ P(\theta|\mathcal{G})\, d\theta \tag{I.41}$$

Equation I.41 marginalizes over an unknown parameter, making it the marginal likelihood of data given the Bayesian structure.

Definition 3.5 (BD Score) *Cooper and Herskovits 1992 proposed the* Bayesian Dirichlet (BD) score*, which uses a Bayesian Perspective to probabilize uncertainties over structures, parameters, and databases. So, it can be defined the same way as the Bayesian Score:*

$$Score_{BD}(\mathcal{G}, D) = P(\mathcal{G}) \int_{\Theta} P(D|\theta, \mathcal{G})\ P(\theta|\mathcal{G})\, d\theta \tag{I.42}$$

The prior probability distributions for the parameter ($P(\theta|\mathcal{G})$) is a Dirichlet *prior here with hyperparameters β_{ijk}, and Heckerman, Geiger, and David M Chickering 1995 considered θ_{ijk} as the multinomial parameter of $P(X_i = k|Pa(X_i) = j)$, and they proved that the equation above is equal to:*

$$Score_{BD}(\mathcal{G}, D) = P(\mathcal{G}) \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\beta_{ij})}{\Gamma(N_{ij} + \beta_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(N_{ijk} + \beta_{ijk})}{\Gamma(\beta_{ijk})} \tag{I.43}$$

where:
r_i : *the number of states of variable* X_i,
$q_i = \prod_{X_i \in Pa(X_i)} r_i$: *the number of states of* $Pa(X_i)$,
j : *index the states of* $Pa(X_i)$,
k : *index the value of* X_i,
N_{ijk} : *the number of samples of D where node* X_i *has the* k^{th} *value, and its parents have the* j_{th} *combination of values,*
$P(X_i = k|Pa(X_i) = j)$*: the probability of* X_i *have the* k^{th} *value given the* j^{th} *state of* $Pa(X_i)$*),*
$\beta_{ij} = \sum_{k=1}^{r_i} \beta_{ijk}$ *and,* $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$,
$\Gamma(.)$: *the gamma function.* ($\Gamma(z) = \int_0^\infty x^{z-1} e^{-x}\, dx$)

In BD, we must manually indicate parameters, which can be difficult. A useful modification is to apply a non-informative prior distribution, such as Jeffreys' prior [Jeffreys 1946], which sets $\beta_{ijk} = 1/2;\ \forall i, j, k$. In line with this approach, Cooper and Herskovits 1992 developed the K2 score, which sets $\beta_{ijk} = 1;\ \forall i, j, k$.

Definition 3.6 (K2 Score) *K2 score is a variant of the BD score when $\beta_{ijk} = 1;\ \forall i, j, k$. The K2 score can be expressed as the following equation, utilizing the definition of gamma function [Cooper and Herskovits 1992]:*

$$Score_{K2}(\mathcal{G}, D) = P(\mathcal{G}) \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}! \tag{I.44}$$

One of the problems with the BD and K2 scores is that they do not always give the same score to equivalent structures. Heckerman, Geiger, and David M Chickering

1995 fixed this problem by proposing the BDe score. They proved that this score has the properties of equivalence and decomposability. They did this by assuming that the score follows the distribution of prior networks $\mathcal{G}$ learned from a dataset with N' (the equivalent sample size, denoted as ESS, represents the user's belief in the prior distribution of the parameters).

Definition 3.7 (BDe Score) Bayesian Dirichlet equivalent (BDe) Score *is the BD score where the prior parameters are defined as following:*

$$N'_{ijk} = N' \; P(X_i = x_k, \; \boldsymbol{Pa}(X_i) = \boldsymbol{pa}(\boldsymbol{x}_j) \mid \mathcal{G}) \tag{I.45}$$

Where N' shows how much the user believes (the weight given) to the prior factor.

$$Score_{BDe}(\mathcal{G}, D) = P(\mathcal{G}) \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(N'_{ij})}{\Gamma(N_{ij} + N'_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(N_{ijk} + N'_{ijk})}{\Gamma(N'_{ijk})} \tag{I.46}$$

where $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$ represents the number of samples of D where node X_i has the kth value, and its parents have the j_{th} combination of values, and $N'_{ij} = \sum_{k=1}^{r_i} N'_{ijk}$ are defined similarly based on prior beliefs of these values. [Heckerman, Geiger, and David M Chickering 1995]

As we discussed, the BDe score addressed the issue of the BD score not being score-equivalent, but it still has some limitations. The primary issue is selecting the appropriate prior distribution for the networks ($P(\mathcal{G})$) and computing the probabilities required for N'_{ijk}, which were not dependent on any specific value of Θ, making them challenging to estimate in practice. To overcome this problem, Buntine 1991 introduced some additional assumptions to the BDe prior, resulting in the creation of the BDeu score.

Definition 3.8 (BDeu Score) Bayesian Dirichlet equivalent uniform (BDeu) Score *is the BDe score with a Dirichlet prior where its hyperparameter is defined as following:*

$$N'_{ijk} = N' \; P(X_i = x_k, \; \boldsymbol{Pa}(X_i) = \boldsymbol{pa}(\boldsymbol{x}_j) \mid \mathcal{G}) = \frac{N'}{r_i q_i} \tag{I.47}$$

Where r_i (resp. q_i) is the domain size of X_i (resp. $\boldsymbol{Pa}(X_i)$). N' is the only free parameter that can be chosen by the user. The equation I.47 concludes that $N'_{ij} = \sum_{k=1}^{r_i} N'_{ijk} = N'/q_i$

$$Score_{BDeu}(\mathcal{G}, D) = P(\mathcal{G}) \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\frac{N'}{q_i})}{\Gamma(N_{ij} + \frac{N'}{q_i})} \prod_{k=1}^{r_i} \frac{\Gamma(N_{ijk} + \frac{N'}{r_i q_i})}{\Gamma(\frac{N'}{r_i q_i})} \tag{I.48}$$

where $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$. [Buntine 1991]

The second category of scoring functions is information-theoretic. The goal is to use information theory to make a model that matches the data D well. One way to measure how good a Bayesian model $\mathscr{B} = (\mathscr{G}, \Theta)$ is for data D is to calculate the logarithm of its likelihood, which is also called log-likelihood, since usually dealing with the logarithm of the likelihood is simpler. This score function is based on the Maximum Likelihood Estimation (MLE) formula given in the Equation I.27. The log-likelihood score is useful in different aspects. First, it shows the expected number of bits needed to encode D optimally, which is useful from an information-theoretic viewpoint. Second, the log-likelihood score is decomposable, which makes it easy to do fast calculations when learning a structure. Also, the log-likelihood is score-equivalent, which is good because I-equivalent models have the same independencies.

Definition 3.9 (Log-likelihood Score) *From the definition of MLE, we have the likelihood of data D given the structure $\mathscr{G}$:*

$$P(D|\mathscr{G}) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \theta_{ijk}^{N_{ijk}} \tag{I.49}$$

The Log-likelihood score *is the logarithm (in base 2) of Equeation I.49:*

$$Score_{LL}(\mathscr{G}, D) = log\ (P(D|\mathscr{G})) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{ijk}\ log\ \theta_{ijk} \tag{I.50}$$

The network learned using the log-likelihood score can only represent conditional independence when the distribution has that independence exactly. However, due to the statistical noise, it is very unlikely that an exact independence will happen. So, in most cases, the network with the maximum likelihood will have too many arcs (or even it would be a complete network). This means that the likelihood score learns a model that overfits the training data by matching the details of the distribution in our training set. To solve this problem, some modifications were applied to the likelihood scores. Some of the most common among these information-theoretic scores are the Akaike Information Criterion (AIC) [Akaike 1970], the Bayesian Information Criterion (BIC) [Schwarz 1978] (also represented as the Minimum Description Length (MDL) [Suzuki 1993, Lam and Bacchus 1994, Suzuki 1999]), the Normalized Maximum Likelihood (NML)[Rissanen 1996], the factorized Normalized Maximum Likelihood (fNML) [Silander, Roos, Kontkanen, et al. 2008], and the quotient Normalized Maximum Likelihood (qNML) [Silander, Leppä-Aho, Jääsaari, et al. 2018]. These scoring functions try to avoid overfitting by preferring simple networks over complex ones. They add a complexity penalty term to the likelihood function to achieve this goal.

The AIC score's penalty term is determined by the number of independent parameters of the model as presented in Definition 3.10. This penalty is not very strong, which means that the AIC score tends to choose models with more independent

parameters than the BIC score that Definition 3.11 represents. The BIC score has a stronger penalty, which makes it prefer models with fewer edges than AIC. The BIC score is commonly used in score-based approaches for Bayesian Network Structure Learning, and it has been shown by Z. Liu, Malone, and Yuan 2012 that this score has a higher speed in finding the correct structure than other information-theoretic scores such as AIC and fNML.

Definition 3.10 (AIC Score) The Akaike Information Criterion (AIC) *is defined by following equation: [Akaike 1970]*

$$Score_{AIC}(\mathcal{G}, D) = Score_{LL}(\mathcal{G}, D) - dim(\mathcal{G}) \tag{I.51}$$

where $dim(\mathcal{G})$ represents the dimension of the model or the number of the network's free parameters:

$$dim(\mathcal{G}) = \sum_{i=1}^{n} (r_i - 1) q_i \tag{I.52}$$

Definition 3.11 (BIC Score) The Bayesian Information Criterion (BIC) *is defined by following equation: [Schwarz 1978]*

$$Score_{BIC}(\mathcal{G}, D) = Score_{LL}(\mathcal{G}, D) - \frac{logN}{2} dim(\mathcal{G}) \tag{I.53}$$

where $dim(\mathcal{G})$ is the same as Equation I.52 and N is the sample size.

A Minimum Description Length (MDL) justification gives the same formula as BIC which is called MDL score. This principle tries to find the simplest model that can compress the data the most (represents the data with the minimum possible description length). [Suzuki 1993, Lam and Bacchus 1994, Suzuki 1999]

The AIC and BIC scores are decomposable and can be considered score-equivalent because the dimensionality penalty, $dim(\mathcal{G})$, is applied consistently across networks that are I-equivalent. However, while both scores aim to balance model fit and complexity, they differ in their ability to identify the correct network as data size increases. BIC is consistent and can find the correct network given a sufficiently large amount of data due to its strong penalty on model complexity. AIC, on the other hand, is not consistent for model selection as it tends to favor more complex models even with large datasets. Both scores may struggle with small data, where the trade-off between fit and complexity is more challenging to balance. Silander, Roos, Kontkanen, et al. 2008 made the factorized Normalized Maximum Likelihood (fNML) score to solve this problem. fNML is based on the Normalised Maximum Likelihood (NML) distribution [Rissanen 1996], which finds the probability of any data set with a specific size for a given graph G. NML score is not decomposable, so Silander, Roos, Kontkanen, et al. 2008 proposed a decomposable version with the cost of missing the score-equivalence property, which you can see in Definition 3.12.

	Bayesian Scores				Information-theoretic Scores					
Score	BD	K2	BDe	BDeu	LL	AIC	BIC	NML	fNML	qNML
Decomposable	*Yes*	*Yes*	*Yes*	*Yes*	*Yes*	*Yes*	*Yes*	*No*	*Yes*	*Yes*
Equivalent	*No*	*No*	*Yes*	*Yes*	*Yes*	*Yes*	*Yes*	*Yes*	*No*	*Yes*
Consistent	*Yes*	*Yes*	*Yes*	*Yes*	*No*	*No*	*Yes*	*Yes*	*Yes*	*Yes*

Table I.4.: Scores Comparison

Definition 3.12 (fNML Score) *The factorized Normalized Maximum Likelihood (fNML) proposed by Silander, Roos, Kontkanen, et al. 2008 as the following equation:*

$$Score_{fNML}(\mathcal{G}, D) = Score_{LL}(\mathcal{G}, D) - \sum_{i=1}^{n} \sum_{j=1}^{q_i} c_{N_{ij}}^{r_i} \tag{I.54}$$

Where $c_{N_{ij}}^{r_i}$ *is the* parametric complexity. *For a multinomial variable that takes r different possible values, it can be computed from the following equation:*

$$c_N^r = \sum_{k_1+k_2+\cdots+k_r=N} \frac{N!}{k_1!k_2!\cdots k_r!} \prod_{j=1}^{r} (\frac{k_j}{N})^{k_j} \tag{I.55}$$

It has been shown in Kontkanen and Myllymäki 2007 this equation can be solved in linear time with a recursive solution:

$$c_N^r = c_N^{r-1} + \frac{N}{r-2} c_N^{r-2} \tag{I.56}$$

Silander, Roos, and Myllymäki 2010 and Z. Liu, Malone, and Yuan 2012 showed that fNML works well on small datasets, but it does not have the equivalency property as we already mentioned. Therefore, Silander, Leppä-Aho, Jääsaari, et al. 2018 made a different score based on NML, called quotient Normalised Maximum Likelihood (qNML), to solve the score-equivalence problem of fNML score.

Table I.4 reviews the scores mentioned in this section and compares their properties. As can be seen, all of the scores are consistent except the log-likelihood score. The NML score is the only score that is not decomposable (among the scores we presented).

3.2.2.2. Approximate Algorithms

Approximate score-based methods are algorithms that use heuristics to search for a structure with a high score in a reasonable time. However, they do not guarantee to

find the optimal structure with the highest score, which means some of these methods can find the best structure with a high probability (when the data is very large), and some can not. This property refers to consistency defined in Definition 3.3. Some approximate score-based methods exhibit high-dimensional consistency, meaning they can effectively identify the best structure even when dealing with a large number of variables, provided there are sufficient samples relative to the complexity of the network. However, their effectiveness may vary depending on the ratio of samples to variables, and they may not discover all the causal relationships in the data.

The **K2 algorithm** by Cooper and Herskovits 1992 is one of the first structure learning algorithms. It is a greedy search algorithm of DAG space that requires a node topological ordering as an input and tries to find the best graph G by looking for the best local parent set $\boldsymbol{Pa}(X_i)$ for each node X_i, using the K2 score (Definition 3.6). The only operator it uses is adding an arc between X_i and a candidate parent from the nodes with higher order. The algorithm begins with an empty graph, and for each node X_i, it iterates over the possible parents X_j with higher order than X_i and picks the one that improves the score the most. It repeats this for each node X_i until no more parents can increase the score. Since K2 does not check all the possible parent sets for each node, it cannot ensure finding the best-scoring DAG for a given topological order.

Remco R Bouckaert 1994 proposed a general **Hill Climbing (HC)** algorithm, which operates a greedy search over the DAG space that does not need a topological order. This algorithm is a simple and popular search strategy. The algorithm may begin from any DAG but usually starts from the empty graph. It iterates over the neighboring DAGs G_n of the current DAG G, obtained by adding, removing, or reversing an arc in G. The score difference for each G_n is stored, and select the structure with the highest difference. If none of the neighboring DAGs improve the score, the algorithm stops at a local (or sometimes global) maximum and returns the DAG as the output.

HC algorithm is very common; however, it has some problems e.g., it may end up with a bad local maximum score. To avoid this problem, different techniques are proposed. Heckerman, Geiger, and David M Chickering 1995 applied local restarts, which means whenever reaches a local maximum, randomly change the DAG and then start hill-climbing from this DAG. Glover 1990 and Remco Ronaldus Bouckaert 1995 used a **tabu list**, which keeps track of the recently visited DAGs and prevents going back to them. It also allows some changes that lower the score to explore new regions that may have a better local maximum. This method helps search for new areas with a higher local maximum. The tabu method may be more sensitive to noise than plain hill climbing.

Model Averaging Hill Climbing (MAHC) [Constantinou, Y. Liu, Neville K Kitson, et al. 2022] is a recent version of hill climbing. It is different from the usual HC because it looks at the average score of a network structure and its valid neighbors, not just the score of the structure itself. Valid neighbors are the ones that can be made by one change in the current structure. The model averaging strategy helps to find better

solutions and not get stuck in local optima. They showed that MAHC is a good and fast way to learn network structures, especially when the data has noise.

The algorithms mentioned above are *search methods over the DAG space*. In contrast, there is another group of methods that *search over the I-equivalence classes*, which means they look for structures with higher scores in the neighboring CPDAGs [David Maxwell Chickering 2002a]. The final output is a DAG obtained from the best CPDAG. As we mentioned when reviewing different scoring functions, many have the score-equivalence property, which searches for equivalent structures more efficiently than searching over DAGs. Making a single change in a DAG can often convert it to an equivalent structure, which can take much time. However, this isn't the case with CPDAGs, as the search space would be much smaller than the DAGs search space. Another benefit of using score with score-equivalent is that search over DAGs is usually stuck in local maxima, which is solved by searching over CPDAGs.

David Maxwell Chickering 2002a, David Maxwell Chickering 2002b proposed one of most common search methods over the equivalence class space, which is called **Greedy Equivalence Search (GES)**. This algorithm consists of two phases: *insert* and *delete* phases. During the insert phase, GES begins with an empty CPDAG. The algorithm then adds edges to the CPDAG one by one, which increases the score the most. During this process, some edges may be changed in direction or orientation according to Meek's rules [Meek 1995] to maintain the CPDAG in the same equivalence class as the optimal DAG. The insert phase stops when it is no longer possible to improve the score by adding edges. In the delete phase, the GES algorithm reverses the process and iteratively deletes edges that increase the score the most. The goal of the delete phase is to remove unnecessary edges from the CPDAG. Like the insert phase, the delete phase stops when reaching the local maxima. Finally, the CPDAG that results from the insert and delete phases is returned as the result. Although GES is an approximate algorithm, under some conditions, it is consistent because it is guaranteed to find a CPDAG that perfectly matches the same conditional independence relationships in the data as the sample size grows to infinity (P-map; Definition 1.24). Joseph Ramsey, M. Glymour, Sanchez-Romero, et al. 2017 proposed an optimized version of GES by parallelizing the computation called **Fast GES (FGES)**. They also made GES more efficient by using some caching and pruning techniques to speed up the computations and reduce the search space.

Another group of approximate algorithms is based on the search methods over the ancestral space. As we saw in Section 3.2.1, most of the causal structure learning algorithms (that take into account the latent variables) are constraint-based. However, Triantafillou and Tsamardinos 2016 proposed a new approach for causal structure learning using score-based methods called the **Global Search for Maximal Ancestral Graphs (GSMAG)** algorithm. They used a greedy search over the MAG space (Definition 2.50) to find the best structure for the continuous data. They used the add, remove, and reverse operators. They made some assumptions to deal with continuous data. They were able to use scoring functions on mixed graphs using residual

iterative conditional fitting to compute MLE for MAGs. In their study, they utilized a BIC score proposed in T. S. Richardson 2009 based on a factorization of the probability distribution for Gaussian MAGs, which is decomposable for each c-component in the MAG, that means recalculation of the score is only needed for the c-components that change when a search operator is used.

3.2.2.3. Exact Algorithms

As we saw in the previous section, the approximate algorithms do not guarantee to return the highest scoring structure that fits the data (except for some that the data grows infinitely, e.g., *GES* [David Maxwell Chickering 2002a]), while the exact algorithms do so even when the data is small. However, we need to mention that returning the structure with the highest score does not mean it will always fit the original structure with high accuracy. This depends on different aspects, like the data properties (i.e., size, noises, missing samples) and the scores used in the algorithms. Unlike the approximate algorithms, which optimize scores locally, exact algorithms guarantee to find the structure that globally maximizes a score. These algorithms often take advantage of different pruning rules for the search space to avoid making redundant searches. These pruning techniques need to be sound to contain the optimal solution in the pruned space. Different approaches for implanting exact algorithms exist, but several can be placed in two groups: dynamic programming and branch and bound.

Dynamic programming is a technique that begins with less complicated sub-problems and saves their results to solve a more complex problem. This technique was one of the first approaches used in exact learning methods proposed by Ott, Imoto, and Miyano 2003 and Koivisto and Sood 2004, which searched in nodes' topological order space. A. P. Singh and Moore 2005 and Silander and Myllymaki 2006 improved the results and reduced the complexity of the previous works by considering that every DAG contains at least one *sink node*: a node that does not have any outgoing arc. Exact algorithms based on dynamic programming need a large amount of memory to be executed, which is one of their main problems[Silander and Myllymaki 2006].

The Branch and Bound (BnB) method begins with a large problem and breaks it down into simpler ones [De Campos, Zeng, and Ji 2009]. The aim of these methods is to find exact solutions while avoiding unnecessary exploration of the search space by discarding irrelevant parts. BnB methods typically store a cache of possible parent sets for each node with their respective score. The cache would have too many items if it contained every parent set, which is not feasible. Therefore, if we need to do an exact search, it would be essential to have pruning rules that can shrink the cache size without losing the guarantee of returning the structure with the highest score[Cussens

2012, Joseph Ramsey, M. Glymour, Sanchez-Romero, et al. 2017, C. P. d. Campos, Scanagatta, Corani, et al. 2018, Correia, Cussens, and C. d. Campos 2020]. Many pruning strategies are specific to a certain score and can drastically reduce the cache size and the search space. De Campos, Zeng, and Ji 2009 and De Campos and Ji 2011 applied this method to find optimal solutions globally. They examined mathematical properties of scores AIC, BIC and BD, to provide upper bounds to eliminate parent sets without evaluating them directly, and they also proposed a search framework that supports the addition of structural constraints such as forced edges, maximum number of parents, and logical operations in addition to the previous types of constraint. They also reported that the cache size was reduced in different magnitudes depending on the network and the score used.

Van Beek and Hoffmann 2015 presented CPBayes, an exact algorithm designed for the specific context of discrete data using constraint programming. Unlike traditional constraint-based methodologies that emphasize conditional independence, CPBayes innovates by focusing on constraints related to the possible parent sets and the inherent acyclicity of Bayesian networks. CPBayes stands out for its use of constraint propagation—a dynamic updating of potential variable values as changes occur within the network. Its model is distinguished by various classes, including nodes' ordering and depth variables, which are governed by a framework of constraints. These include symmetry-breaking constraints that discard redundant solutions, acyclicity constraints that uphold the non-circular nature of the network, and dominance constraints for effective pruning. As a depth-first branch-and-bound (BnB) algorithm, CPBayes navigates the permutation space of node sequences, utilizing score bounds and the aforementioned constraints to systematically eliminate non-viable solutions, thus optimizing the search for the most suitable Bayesian network structure.

3.2.3. Hybrid Algorithms

As mentioned in previous sections, constraint-based approaches may not always produce precise results due to errors caused by statistical tests, although they make constraint-based methods relatively fast and deterministic. However, they depend on a chosen significance level α to judge the independence of two variables. This can lead to instability in the learning process, meaning that wrong decisions can cause more errors to accumulate in the final structure. On the other hand, score-based approaches often find suboptimal solutions as they explore vast search spaces; hence, they are more robust than constraint-based algorithms in terms of errors, but their search process usually makes them slower. Hybrid algorithms have been developed to take advantage of the strengths of both approaches. These algorithms combine constraint-based and score-based approaches, proposing the best properties of both. The most common method of combining these approaches involves using a constraint-based approach to *restrict* the search space or quickly find an initial graph. Subsequently, a score-based approach is employed to locate a graph with a *maximum* score. Neville Kenneth Kitson, Constantinou, Guo, et al. 2023 called this group *Restrict/Maximize*

Algorithm	Year	Search Space	Restrict	Maximize	Output
CB	1993	DAG	PC	K2	DAG
SC	1999	DAG	MII	Tabu	DAG
MMHC	2006	DAG	MMPC	Tabu	DAG
COS	2008	DAG	MMPC	Exact Algorithm	DAG
PCB	2011	DAG	Partial Correlation CI	HC	DAG
ARGES	2018	Equivalence	MMPC	GES	CPDAG
MMMHC	2018	Ancestral	MMPC	GSMAG	MAG
CCHM	2020	Ancestral	CFCI	GHC	MAG
HMCMC	2022	Ordering	PC	MCMC	DAG

Table I.5.: Hybrid Structure Learning Algorithms

algorithms and divided them into four subgroups based on their search space: DAG, equivalence class, node ordering, and ancestral graphs. We will review in this section some of the most popular methods of each of these groups, which are also listed in Table I.5 in order of their years. Since most Hybrid algorithms employ the search in DAG space, we begin with this group.

One of the earliest hybrid approaches is Constraint-Bayesian (**CB**), proposed by M. Singh and Valtorta 1993. For the restrict phase, they used the PC (Section 3.2.1.1) algorithm that produces the CPDAG. In the maximize phase, first, they orient the undirected edges in a direction that maximizes the K2 score (Definition 3.6) of the endpoints. Then, they use the K2 algorithm (Section 3.2.2.2) to find the DAG with the maximum score. Friedman, Nachman, and Peér 1999 proposed another similar approach, called the Sparse Candidate (**SC**), which made use of the Mutual Information Independence (MII) tests in the restrict phase for finding the nominated parents of each node. In the maximize phase, the tabu algorithm (Section 3.2.2.2) is employed, constrained by those parents. One of the most popular hybrid algorithms over the DAG space is Max-Min Hill Climbing (**MMHC**) by Tsamardinos, Brown, and Aliferis 2006. They utilized the MMPC (Section 3.2.1.1) as the constraint-based approach to finding the skeleton. Then, for the score-based method, they took advantage of the tabu algorithm, which only used the edges from the skeleton of the restrict phase to learn the DAG.

All the mentioned approaches exploited the approximate algorithm for their maximize phase, while Perrier, Imoto, and Miyano 2008 used an exact score-based dynamic programming algorithm (Section 3.2.2.3) in their hybrid method called Constrained Optimal Search (**COS**). Their restrict phase is like MMHC, which means they employ the MMPC algorithm to find the skeleton. However, they use a high CI significance level, which makes their skeleton denser than MMHC, to have higher certainty for finding the globally optimal graph in the restrict phase.

The hybrid methods named above are developed for discrete variables. One of these methods proposed for continuous variables is called Partial Correlation Bayesian (**PCB**) by J. Yang, Li, and Wang 2011. As we can see in the method name, they utilized partial correlation CI tests to find the skeleton in the restrict phase and a hill-climbing (Section 3.2.2.2) method for the maximize phase.

One of the restrict/maximize methods for equivalence class search space is Adaptively Restricted Greedy Equivalence Search (**ARGES**), which is developed for continuous variables. In the restrict phase, they adopted MMPC, followed by a modified GES for the maximize phase (Section 3.2.2.2).

Another group of hybrid methods is in the node ordering space; for instance, **Hybrid MCMC** is proposed by Kuipers, Suter, and Moffa 2022. They used PC (Section 3.2.1.1) to restrict the search space. They utilized the Markov Chain Monte Carlo (MCMC) for maximizing the score, either in the ordering space (using Order-MCMC by Friedman and Koller 2003) or in the partition space (using the Partition-MCMC by Kuipers and Moffa 2017).

Tsirlis, Lagani, Triantafillou, et al. 2018 proposed the MAG Max-Min Hill Climbing (**MMMHC**) method that combined the MMPC as the restricting part and GSMAG (Section 3.2.2.2) as the maximizing phase, which means they are causal insufficient and let them learn the structure with the presence of the latent variable. Another method in the ancestral graph (Definition 2.48) is proposed by Chobtham and Constantinou 2020 called **CCHM**, which combines CFCI (constraint-based, Section 3.2.1.2) that creates the MAGs (Definition 2.50) and greedy hill-climbing (score-based, Section 3.2.2.2) algorithms to orient the edges using the BIC score (Definition 3.11).

II. Contributions

Sommaire

4. A New Full Score-based Causal Learning Algorithm . . . 81
   4.1. Triangles in DAGs . . . 81
      4.1.1. Genuine vs. Latent Triangles . . . 91
   4.2. The Algorithm . . . 93
      4.2.1. Triangles Discrimination . . . 93
      4.2.2. Finding d-Separating Sets . . . 96
      4.2.3. Modifying the Structure . . . 100
      4.2.4. Complexity Analysis . . . 102
   4.3. Experiments . . . 104
   4.4. Conclusion . . . 116
5. A Causal Learning Approach for Medical Data . . . 117
   5.1. Introduction . . . 117
   5.2. Methodology . . . 118
      5.2.1. Data Description . . . 118
      5.2.2. Handling Missing Data . . . 120
      5.2.3. Bootstrapping for Data Augmentation . . . 122
      5.2.4. Constructing a Trustworthy Structure . . . 124
      5.2.5. Weighted Causal Effects . . . 126
      5.2.6. Latent Variables Identification . . . 132
   5.3. Experiments . . . 132
      5.3.1. Model Generation . . . 132
      5.3.2. Validation of Bootstrapping Technique . . . 133
      5.3.3. Bootstrapping and Structure Learning . . . 141
      5.3.4. Results . . . 141
      5.3.5. Discussion . . . 149
   5.4. Conclusion . . . 149

In Part I, we delved into the intricacies of Causal and Bayesian networks, uncovering their foundational similarities and distinct differences. We discovered the critical role of latent variables, which, though invisible in our datasets, are key players in the causal framework. Traditionally, the field has leaned towards constraint-based methods for unraveling these networks, partly due to skepticism towards score-based algorithms' ability to handle latent variables effectively. This skepticism is rooted in two main challenges: difficulty of quantifying occurrences of variable states for score

computation when some variables remain unseen and reliance on Directed Acyclic Graphs (DAGs), whose structure offers limited insight into latent variables.

In response to these challenges, the scientific community has explored alternative graphical structures like CPDAGs (Definition 2.21), which provide a richer depiction of node interconnections, thereby offering a more conducive environment for latent variable analysis. Furthermore, the advent of Maximal Ancestral Graphs (MAGs) (Definition 2.50) has introduced a robust framework for structure learning in the presence of latent variables, albeit with a focus on continuous variables. This backdrop highlights a notable void: a lack of score-based learning strategies adept at managing discrete variables, underscoring the limitations of current methodologies.

This part aims to fill this void by introducing a novel score-based approach we have developed, specifically designed to identify latent variables within networks from observable discrete data. Our method challenges the conventional belief that score-based algorithms are ill-equipped for this task, showcasing the feasibility and efficacy of such an approach in the discrete variable domain. In Section 4, we will unveil this method, outlining the principles of our causal structure learning algorithm and its capability to pinpoint latent variables. This contribution not only contests the prevailing assumptions within the field but also broadens the scope of causal inference research, enhancing our ability to decipher causal mechanisms across various datasets.

4. A New Full Score-based Causal Learning Algorithm

In Section 3.2 of the previous part, we discussed the limitations of constraint-based algorithms, particularly their sensitivity to statistical errors when dealing with latent variables. On the other hand, score-based algorithms have shown greater robustness in scenarios with no latent variables. This contrast has spurred our interest in leveraging the strengths of score-based approaches to address the challenges posed by latent variables in structure learning. This section unveils our novel algorithm, designed without the need for prior knowledge about the latent variables or their quantity. We demonstrate how DAGs, derived from observational data through score-based methods, can encapsulate essential information for uncovering numerous latent variables and pinpointing their exact locations.

Our algorithm differs from existing approaches by utilizing:

- **Score Dependency**: It exclusively utilizes scoring metrics, harnessing their robustness to statistical errors and their efficacy in evaluating model fit.
- **DAG Utilization**: By employing DAGs, our method takes advantage of their structural simplicity and interpretability instead of using other kinds of structures like CPDAGs and MAGs, facilitating a clearer understanding of causal relationships.
- **Latent Variable Identification**: It is adept at not only identifying the presence of latent variables but also precisely locating them within the network, enhancing the depth and accuracy of causal inference.

We will begin by elucidating some key concepts (Section 4.1) that lay the groundwork for our algorithm. These concepts are instrumental in grasping the details of our approach and its innovative handling of latent variables. Following this foundational discussion, we delve into the details of our algorithm (Section 4.2), highlighting its design and operational principles. Subsequent experiments (Section 4.3)) are then presented to validate our method's effectiveness, ending in a reflective conclusion (Section 4.4) that encapsulates our findings and their implications for the field of causal learning.

4.1. Triangles in DAGs

In this section, we challenge a common assumption in the field of causal inference by demonstrating that Directed Acyclic Graphs (DAGs) possess the inherent capacity to reveal latent variables through their structural configurations. Central to our argument is the concept of 'triangles' within DAGs, which we define as specific arrangements formed by three interconnected nodes. Unlike in undirected graphs, where a triangle simply denotes three mutually connected nodes, the directional

nature of DAGs introduces complexity by allowing for various configurations of these connections. However, the acyclic property of DAGs necessitates the exclusion of any configurations that would result in directed cycles, thus limiting the possible triangle formations. Out of the eight theoretically conceivable triangle configurations, two are inherently incompatible with the acyclic nature of DAGs, leaving us with six viable structures. These permissible configurations are crucial for our analysis, as they serve as the foundation for explaining the relationship between triangular formations and the underlying presence of latent variables. Figure II.1 visually represents these six configurations, providing a clear depiction of the potential structures through which latent variables can influence observable relationships within a DAG.

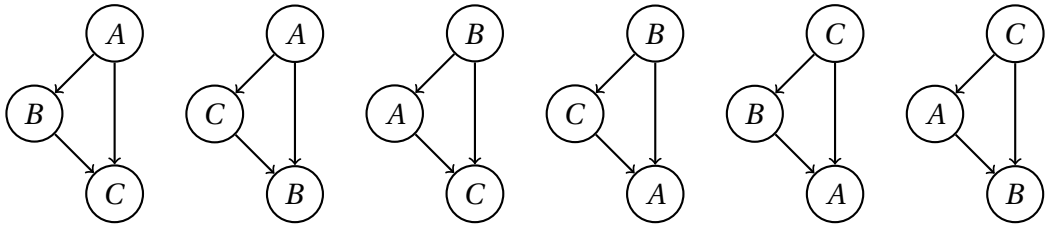


Figure II.1.: Triangles in a DAG

In the field of structure learning, our focus is particularly on scenarios where latent variables play an important role, yet remain hidden within the dataset at our disposal. This dataset, denoted as $\mathscr{D}$, is essentially a reduced projection of a more comprehensive dataset, $\mathscr{D}^*$, which contains a broader spectrum of variables, both observable ($\mathbf{X}_O$) and hidden ($\mathbf{X}_H$). The derivation of $\mathscr{D}$ from $\mathscr{D}^*$ involves the exclusion of hidden variables, resulting in a dataset confined to observable variables alone. This process is based on the assumption that $\mathscr{D}^*$ arises from a probability distribution P^* over the entire set of variables $\mathbf{X} = \mathbf{X}_O \bigcup \mathbf{X}_H$, and this distribution is decomposable according to a Directed Acyclic Graph (DAG), symbolized as $\mathscr{G}^*$. Our attempt is not merely to analyze $\mathscr{D}$ but to utilize it as a lens through which we can reconstruct the latent structure of $\mathscr{G}^*$. Within this context, the configurations of triangles within DAGs emerge as a critical analytical tool. These configurations, by their structural properties, offer valuable insights into the interplay between observable and hidden variables, thus facilitating the complicated process of inferring the original graph $\mathscr{G}^*$ from the observable data encapsulated in $\mathscr{D}$.

Based on the foundational concepts introduced in Section 3.2.2 of the preceding part, we move deeper into the mechanics of score-based algorithms within the context of latent variable detection. These algorithms, guided by the principle of score optimization, strategically introduce arcs to enhance the model's fit to the data. An illustrative example of this process is depicted using a semi-Markovian model, described in Definition 2.4 and visually represented in Figure II.2. Here, the original structure, $\mathscr{G}^*$, comprises a set of observable variables, $\mathbf{X}_O = \{A, B, C, D\}$, and a hidden variable, $\mathbf{X}_H = \{L\}$.

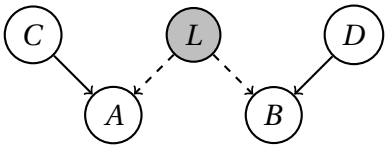
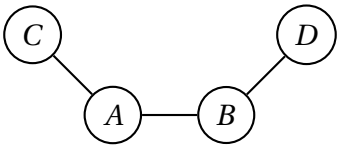
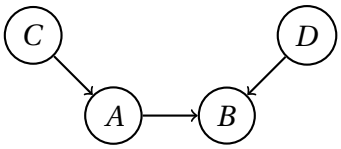


Figure II.2.: Original BN ($\mathcal{G}^*$)

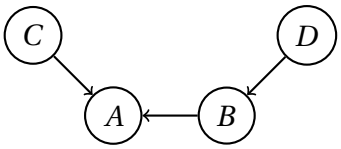
The score-based algorithm would add arcs between the nodes C—A and D—B since they are d-connected in $\mathcal{G}^*$ and adding them in $\mathcal{G}$ would increase the score significantly. Nodes A and B cannot be d-separated given any set of nodes in $\mathcal{G}^*$. The path between them $\langle A, L, B\rangle$ is always active since node L is not observable. So, the score-based algorithm would also add an arc between A and B. Figure II.3a shows the constructed skeleton. To check how these edges will be oriented, we need to employ more information from $\mathcal{G}^*$. In this graph, nodes C and B (resp. D and A) are d-connected given $\mathbf{Z} = \{A\}$ (resp. $\mathbf{Z} = \{B\}$), hence the score-based algorithm needs to orient the triple $C \longrightarrow A \longleftarrow B$ (resp. $D \longrightarrow B \longleftarrow A$) as a v-structure to obtain a higher score. Clearly, it is not possible to have both of these orientations since we will have $A \longrightarrow B$ (Figure II.3b) and $A \longleftarrow B$ (Figure II.3c) in the same graph, and this will create a cycle which is forbidden in a DAG. So, the algorithm has to choose one of these orientations. As we mentioned in the original graph, nodes C and B (resp. D and A) are d-connected given $\mathbf{Z} = \{A\}$ (resp. $\mathbf{Z} = \{B\}$), but in Figure II.3b (resp. Figure II.3c) this condition cannot be satisfied. Therefore, the algorithm will add an arc from $C \longrightarrow B$ (resp. $D \longrightarrow A$) to make C and B (resp. D and A) dependant given A (resp. B), as it is shown in Figure II.3d (resp. Figure II.3e). The reason for orienting these



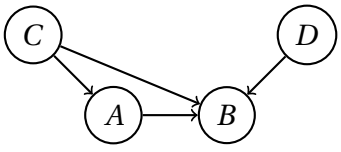
(a) Primary Skeleton



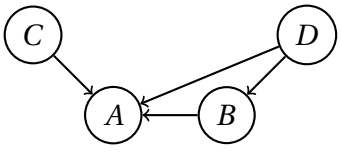
(b) Possible Orientation 1



(c) Possible Orientation 2



(d) Possible Learned Structure 1 ($\mathcal{G}$)



(e) Possible Learned Structure 2 ($\mathcal{G}$)

Figure II.3.: Creation of a Triangle in a Score-based Learning Algorithm

two arcs in the mentioned directions is that nodes C and D are d-separated given $\mathbf{Z} = \{A, B\}$ in $\mathcal{G}^*$ and these directions will satisfy the same condition in $\mathcal{G}$ which in results in increasing the score.

The figures II.2 and II.3 clearly demonstrate how a latent variable in a semi-Markovian model leads to the creation of a triangle pattern in the structure learned by a score-based algorithm. This example lays the groundwork for our next steps, where we will delve into the specifics of this relationship through Proposition 4.1. This proposition aims to formally link the presence of latent variables in the original graph $\mathcal{G}^*$ with the triangles that appear in the learned graph $\mathcal{G}$. Before we get to that, we need to understand what it means when the learned graph $\mathcal{G}$ does not show a direct connection between two nodes, which will be explained in the following lemma.

Lemma 4.1 *Let $\mathbf{X}_O$ and $\mathbf{X}_H$ be two disjoint sets of random variables and let $\mathbf{X} = \mathbf{X}_O \cup \mathbf{X}_H$. Let $\mathcal{D}^*$ be a dataset generated by some distribution P^* over $\mathbf{X}$ for which there exists a perfect map $\mathcal{G}^* = (\mathbf{X}, \mathcal{E})$, and let $\mathcal{D}$ be the projection of $\mathcal{D}^*$ over $\mathbf{X}_O$, i.e., the dataset resulting from the removal from $\mathcal{D}^*$ of all the values of the variables of $\mathbf{X}_H$. Let $\mathcal{G}$ be a DAG maximizing the BIC score over $\mathcal{D}$. Then, as $|\mathcal{D}| \longrightarrow \infty$, if, in Graph $\mathcal{G}$, there is no arc between a pair of node (A, B) and no directed path (Definition 1.7) from B to A, then $A \perp\!\!\!\perp_{P^*} B | \boldsymbol{Pa}_{\mathcal{G}}(B)$.*

Proof of Lemma 4.1. As there is no directed path from B to A in $\mathcal{G}$, adding $A \longrightarrow B$ cannot create a directed cycle hence this would produce a new DAG. If $\mathcal{G}$ does not contain this arc, this therefore means that the BIC score $S(B|\mathbf{Pa}_{\mathcal{G}}(B) \cup \{A\})$ is lower than or equal to $S(B|\mathbf{Pa}_{\mathcal{G}}(B))$. $\mathcal{D}$ is the projection of $\mathcal{D}^*$ over $\mathbf{X}_O$, so the distribution P which generated $\mathcal{D}$ is the projection of P^* over $\mathbf{X}_O$, hence, for any $\mathbf{Z}, \mathbf{W} \subseteq \mathbf{X}_O$, $P(\mathbf{Z}|\mathbf{W}) = P^*(\mathbf{Z}|\mathbf{W})$. As a consequence, as $|\mathcal{D}| \longrightarrow \infty$, the BIC score can be expressed in terms of mutual information ($\mathbf{I}_{P^*}$) and entropy ($\mathbf{H}_{P^*}$) over P^* [Koller and Friedman 2009]:

$$S(B|\mathbf{Pa}_{\mathcal{G}}(B) \cup \{A\}) \longrightarrow |\mathcal{D}| \times [\mathbf{I}_{P^*}(B; \mathbf{Pa}_{\mathcal{G}}(B) \cup \{A\}) - \mathbf{H}_{P^*}(B)] - \frac{\log|\mathcal{D}|}{2} dim(B|\mathbf{Pa}_{\mathcal{G}}(B) \cup \{A\}),$$

where $dim(B|\mathbf{Pa}_{\mathcal{G}}(B) \cup \{A\}) = (|\Omega_B| - 1) \times |\Omega_A| \times \prod_{X \in \mathbf{Pa}_{\mathcal{G}}(B)} |\Omega_X|$. So, if α denotes the difference between the two scores, we have that:

$$\begin{aligned} \alpha &= S(B|\mathbf{Pa}_{\mathcal{G}}(B) \cup \{A\}) - S(B|\mathbf{Pa}_{\mathcal{G}}(B)) \\ &= |\mathcal{D}| \times [\mathbf{I}_{P^*}(B; \mathbf{Pa}_{\mathcal{G}}(B) \cup \{A\}) - \mathbf{I}_{P^*}(B; \mathbf{Pa}_{\mathcal{G}}(B))] - \delta \log|\mathcal{D}|/2, \end{aligned}$$

with $\delta = dim(B|\mathbf{Pa}_{\mathcal{G}}(B) \cup \{A\}) - dim(B|\mathbf{Pa}_{\mathcal{G}}(B))$. When $|\mathcal{D}| \longrightarrow \infty$, $\log|\mathcal{D}|$ is infinitely smaller than $\mathcal{D}$. Hence $\alpha \leq 0$ if and only if $\mathbf{I}_{P^*}(B; \mathbf{Pa}_{\mathcal{G}}(B) \cup \{A\}) \leq \mathbf{I}_{P^*}(B; \mathbf{Pa}_{\mathcal{G}}(B))$. But, for every probability distribution Q and every $X, Y, \mathbf{Z}$, it always holds that $\mathbf{I}_Q(X; Y, \mathbf{Z}) \geq \mathbf{I}_Q(X; \mathbf{Z})$, with equality only if $X \perp\!\!\!\perp_Q Y | \mathbf{Z}$. Hence $\alpha \leq 0$ if and only if $\mathbf{I}_{P^*}(B; \mathbf{Pa}_{\mathcal{G}}(B) \cup \{A\}) = \mathbf{I}_{P^*}(B; \mathbf{Pa}_{\mathcal{G}}(B))$. This implies that $A \perp\!\!\!\perp_{P^*} B | \mathbf{Pa}_{\mathcal{G}}(B)$. ■

In the context of structure learning with observable and latent variables, Lemma 4.1 posits a foundational principle: when two observable variables exhibit no direct

connection within a learned structure (DAG), and no directed path exists from second variable to the first one, it can be inferred that these variables are conditionally independent, given the set of parents of the second variable. This conditional independence is crucial for understanding the underlying causal structure in the presence of latent variables, as it indicates what it means to lack direct causal influence between the two observable variables within the confines of the learned structure.

One outcome of this lemma is that whenever there is an arc between two observable nodes in the generating graph ($\mathcal{G}^*$), then there would be either the same arc or the reversed arc in the learned structure ($\mathcal{G}$).

Corollary 4.1 *Let $\mathbf{X}_O$ and $\mathbf{X}_H$ be two disjoint sets of random variables and let $\mathbf{X} = \mathbf{X}_O \cup \mathbf{X}_H$. Let $\mathcal{D}^*$ be a dataset generated by some distribution P^* over $\mathbf{X}$ for which there exists a perfect map $\mathcal{G}^* = (\mathbf{X}, \mathcal{E})$, and let $\mathcal{D}$ be the projection of $\mathcal{D}^*$ over $\mathbf{X}_O$. Let $\mathcal{G}$ be a DAG maximizing the BIC score over $\mathcal{D}$. Then, as $|\mathcal{D}| \longrightarrow \infty$, for any $A, B \in \mathbf{X}_O$, if $\mathcal{G}^*$ contains Arc $A \longrightarrow B$, then Graph $\mathcal{G}$ contains either Arc $A \longrightarrow B$ or $B \longrightarrow A$.*

Proof of Corollary 4.1. Assume that, for some pair $A, B \in \mathbf{X}_O$ such that there exists an arc between A and B in $\mathcal{G}^*$, Graph $\mathcal{G}$ contains neither Arc $A \longrightarrow B$ nor Arc $B \longrightarrow A$. It is impossible that $\mathcal{G}$ contains both a directed path from A to B and another one from B to A because the concatenation of these paths would be a directed cycle. Without loss of generality, assume that there exists no directed path from B to A in $\mathcal{G}$, then, by Lemma 4.1, $A \perp\!\!\!\perp_{P^*} B | \mathbf{Pa}_{\mathcal{G}}(B)$. As $\mathcal{G}^*$ is a perfect map, this implies that $\langle A \perp_{\mathcal{G}^*} B | \mathbf{Pa}_{\mathcal{G}}(\mathbf{B}) \rangle$, a contradiction since $\mathcal{G}^*$ contains an arc between A and B. ■

Corollary 4.1 extends Lemma 4.1's implications by addressing scenarios where a direct connection between two observable variables is present in the learned structure. It asserts that if an arc exists between two observable variables in the original graph, the learned graph will retain a connection between these variables, albeit potentially in the reversed direction. This persistence of connectivity underscores the robustness of the learning algorithm in preserving relationships between observable variables and finding a correct skeleton. Lemma 4.1 and its following corollary help us to formalize what we explained in the example at the beginning.

Proposition 4.1 *Assume that there exists a perfect map $\mathcal{G}^* = (\mathbf{X}, \mathcal{E})$ for Distribution P^* and that $\mathbf{X}_H$ is the set of latent confounders, i.e., all the nodes of $\mathbf{X}_H$ have no parent and exactly two children in $\mathcal{G}^*$, which both belong to $\mathbf{X}_O$.*

Let $L \in \mathbf{X}_H$ be any variable such that both of its children A, B in $\mathcal{G}^*$ have at least one parent in $\mathbf{X}_O$. Then, if $\mathcal{G}$ is a DAG maximizing the BIC score over $\mathcal{D}$, as $|\mathcal{D}| \longrightarrow \infty$, $\mathcal{G}$ contains an arc between A and B. Without loss of generality, assume this is Arc $A \longrightarrow B$. Then, for every $C \in \boldsymbol{Pa}_{\mathcal{G}^*}(A) \cap \mathbf{X}_O$, (A, B, C) is a clique in $\mathcal{G}$.

Proof of Proposition 4.1. Let L be a variable in $\mathbf{X}_H$ and let A, B be its children. Let $\mathcal{G}$ be a graph maximizing the BIC score over $\mathcal{D}$. First, let us prove that there must exist an arc between A and B in $\mathcal{G}$. Assume the contrary. It is impossible that there

exists both a directed path $\mathscr{C}_{AB}$ from A to B in $\mathscr{G}$ and a directed path $\mathscr{C}_{BA}$ from B to A in $\mathscr{G}$ because their concatenation would be a directed cycle, a contradiction since $\mathscr{G}$ is a DAG. Without loss of generality, assume that there exists no directed path from B to A in $\mathscr{G}$. Then, as, by hypothesis, there exists no arc between A and B, by Lemma 4.1, $A \perp\!\!\!\perp_{P^*} B | \mathbf{Pa}_{\mathscr{G}}(B)$. But this is impossible because, whatever the set $\mathbf{Z} \subseteq \mathbf{X}_O \backslash \{A, B\}$, path $\langle A, L, B \rangle$ is active given $\mathbf{Z}$ in $\mathscr{G}^*$, so that $\langle A \not\perp\!\!\!\perp_{\mathscr{G}^*} B | \mathbf{Pa}_{\mathscr{G}}(\mathbf{B}) \rangle$, which contradicts $A \perp\!\!\!\perp_{P^*} B | \mathbf{Pa}_{\mathscr{G}}(B)$ since $\mathscr{G}^*$ is a perfect map. So, there must exist an arc between A and B in $\mathscr{G}$. In the rest of the proof, without loss of generality, assume that it contains Arc $A \longrightarrow B$.

Let $C \in \mathbf{Pa}_{\mathscr{G}^*}(A) \cap \mathbf{X}_O$. Such a C exists by the hypotheses at the beginning of the Proposition. By Corollary 4.1, $\mathscr{G}$ contains an arc between A and C. As for an arc between B and C, assume that there exists none in $\mathscr{G}$. Note that path $\mathscr{C}_{CB} = \langle C, A, L, B \rangle$ is always active in $\mathscr{G}^*$ given any $\mathbf{Z} \subseteq \mathbf{X}_O \backslash \{B, C\}$ that contains A. So $B \not\perp\!\!\!\perp_{P^*} C | \mathbf{Z}$ since $\mathscr{G}^*$ is a perfect map. Now, as shown at the beginning of the proof, there exists either i) no directed path from B to C in $\mathscr{G}$; or ii) no directed path from C to B in $\mathscr{G}$. If Case i) obtains, by Lemma 4.1, $C \perp\!\!\!\perp_{P^*} B | \mathbf{Pa}_{\mathscr{G}}(B)$, which is impossible since $A \in \mathbf{Pa}_{\mathscr{G}}(B)$ (according to the preceding paragraph), so that path $\mathscr{C}_{CB}$ is active in $\mathscr{G}^*$ given $\mathbf{Z} = \mathbf{Pa}_{\mathscr{G}}(B)$, a contradiction (since it is equivalent to $C \not\perp\!\!\!\perp_{P^*} B | \mathbf{Pa}_{\mathscr{G}}(B)$). If Case ii) obtains (but not Case i)), then the arc added previously between A and C could not be $C \longrightarrow A$ because, since Case i) does not obtain, there exists a directed path from B to C, hence also a directed path from A to C (since $\mathscr{G}$ contains Arc $A \longrightarrow B$). So Arc $C \longrightarrow A$ would create a directed cycle. Hence, $A \in \mathbf{Pa}_{\mathscr{G}}(C)$. Now, by Lemma 4.1, $B \perp\!\!\!\perp_{P^*} C | \mathbf{Pa}_{\mathscr{G}}(C)$, which is impossible since $\mathbf{Z} = \mathbf{Pa}_{\mathscr{G}}(C) \supseteq \{A\}$ would make path $\mathscr{C}_{CB}$ active in $\mathscr{G}^*$, a contradiction. As a consequence, there must necessarily exist an arc between B and C in $\mathscr{G}$. So, overall, $\mathscr{G}$ contains clique (A, B, C). ■

This proposition elaborates on the specific case where a latent variable influences two observable variables, both of which are also influenced by at least one other observable variable as shown in Figure II.2. It establishes that, under these conditions, the learned DAG will invariably include a direct arc between the two observable variables influenced by the latent variable. Furthermore, if another observable variable exerts influence on one of the pair, the trio will form a clique in the learned structure. This configuration is indicative of the latent variable's influence which creates a triangle in the learned structure and is pivotal for the identification and inference of latent structures within the observable data.

Proposition 4.1 also emphasizes that whenever node L is the only parent of A (resp. B), no learning algorithm can distinguish between a graph $\mathscr{G}^*$ that has a latent variable L with two separate children A, B and a graph $\mathscr{G}^*$ without latent variable L but with an arc $A \longrightarrow B$ (resp. $B \longrightarrow A$). In other words, no triangle would be created if one of these children has no observable parent. The following proposition presents this *indistinguishability* property.

Proposition 4.2 *Let $\mathcal{B}^* = (\mathcal{G}^*, \Theta)$ be a Bayesian network, with $\mathcal{G}^* = (\mathbf{X}, \mathcal{E})$. Let $L \in \mathbf{X}_H$ be such that $\boldsymbol{Ch}_{\mathcal{G}^*}(L) = \{A, B\} \subseteq \mathbf{X}_O$ and $\boldsymbol{Pa}_{\mathcal{G}^*}(L) = \emptyset$. In addition, assume that $\boldsymbol{Pa}_{\mathcal{G}^*}(A) = \{L\}$. Finally, let $\mathcal{G}$ be the graph resulting from the removal of L (and its outgoing arcs) from $\mathcal{G}^*$ and the addition of arc $A \longrightarrow B$ (if $\mathcal{G}^*$ does not already contain it). Then, for any triple of disjoint subsets of variables $\mathbf{U}, \mathbf{V}, \mathbf{W}$ of $\mathbf{X}_O$, we have that:*

$$\langle \mathbf{U} \perp_{\mathcal{G}} \mathbf{V} | \mathbf{W} \rangle \iff \langle \mathbf{U} \perp_{\mathcal{G}^*} \mathbf{V} | \mathbf{W} \rangle. \tag{II.1}$$

Proof of Proposition 4.2. First, note that $\mathcal{G}^*$ does not contain Arc $B \longrightarrow A$, otherwise $\mathbf{Pa}_{\mathcal{G}^*}(A)$ would contain B. Adding Arc $A \longrightarrow B$ cannot create a directed cycle in $\mathcal{G}^*$ because A has only one parent, L, which cannot be involved in any cycle since it has no parent. $\langle \mathbf{U} \perp_{\mathcal{G}} \mathbf{V} | \mathbf{W} \rangle$ is equivalent to $\langle U \perp_{\mathcal{G}} V | \mathbf{W} \rangle$ for all $U \in \mathbf{U}$ and $V \in \mathbf{V}$. So, we just need to prove that, for any $U, V \in \mathbf{X}_O$, $U \neq V$, $\langle U \perp_{\mathcal{G}} V | \mathbf{W} \rangle \iff \langle U \perp_{\mathcal{G}^*} V | \mathbf{W} \rangle$. Let $\mathbf{Desc}_{\mathcal{G}}(A)$ denote the set of descendants of A in $\mathcal{G}$, i.e., $\mathbf{Desc}_{\mathcal{G}}(A) = \{X :$ there exists a directed path from A to $X\}$. Note first that $\mathbf{Desc}_{\mathcal{G}}(A) = \mathbf{Desc}_{\mathcal{G}^*}(A) \cup \mathbf{Desc}_{\mathcal{G}^*}(B)$ and, for all $X \in \mathbf{X} \backslash \{A, L\}$, we have that $\mathbf{Desc}_{\mathcal{G}}(X) = \mathbf{Desc}_{\mathcal{G}^*}(X)$.

Now, consider any simple path $\mathcal{C}^* = \langle X_1 = U, \ldots, X_k = V \rangle$ between U and V in $\mathcal{G}^*$. It is blocked in $\mathcal{G}^*$ if and only if i) it contains a convergent connection at some node T such that neither T nor its descendants are in $\mathbf{W}$; or ii) it contains a non-convergent connection at some node $T \in \mathbf{W}$.

Assume path $\mathcal{C}^*$ does not contain L, then it also exists in $\mathcal{G}$. Case i) cannot occur at $T = A$ because A has only one parent in $\mathcal{G}^*$ or at $T = L$ because $\mathcal{C}^*$ does not contain L by hypothesis. Since all the other nodes in $\mathcal{G}^*$ have the same set of descendants as their counterparts in $\mathcal{G}$, the convergent connections in $\mathcal{C}^*$ have therefore exactly the same status (blocked,active) in $\mathcal{G}$ and $\mathcal{G}^*$. As for case ii), any non-convergent connection in $\mathcal{C}^*$ is the same in $\mathcal{G}$ and $\mathcal{G}^*$ and cannot involve L (by hypothesis). Hence, they have the same status in $\mathcal{G}$ and $\mathcal{G}^*$. So $\mathcal{C}^*$ is blocked or active in both $\mathcal{G}$ and $\mathcal{G}^*$.

Now, if $\mathcal{C}^*$ contains L, it includes the non-convergent connection $A \longleftarrow L \longrightarrow B$. Let $\mathcal{C}$ be the path of $\mathcal{G}$ obtained from $\mathcal{C}^*$ by substituting $A \longleftarrow L \longrightarrow B$ by $A \longrightarrow B$. For the same reasons as above, all the nodes different from A, L have the same connections in $\mathcal{C}$ and $\mathcal{C}^*$ and the same set of descendants. Hence their status (blocked, active) are the same in $\mathcal{C}$ and $\mathcal{C}^*$. Node A cannot have a convergent connection in $\mathcal{C}^*$ because $\mathbf{Pa}_{\mathcal{G}^*}(A) = \{L\}$. In $\mathcal{C}$, its connection is also non-convergent since its child B in $\mathcal{G}$ is its neighbor in $\mathcal{C}$. So, the connection at A is non-convergent and has the same status in both $\mathcal{C}$ and $\mathcal{C}^*$. Node L has a non-convergent connection in $\mathcal{C}^*$ but, as it is unobserved, it cannot belong to $\mathbf{W}$ and cannot block the path. So, removing it from $\mathcal{C}^*$ cannot change the status of the path. Hence, overall, all the paths in $\mathcal{G}^*$ can be mapped into a path in $\mathcal{G}$ with exactly the same status.

Conversely, let $\mathscr{C}$ be a simple path in $\mathscr{G}$. If this path does not include Arc $A \longrightarrow B$, then it also belongs to $\mathscr{G}^*$ and the same reasoning as above shows that this path has the same status in $\mathscr{G}$ and $\mathscr{G}^*$.

If, on the other hand, $\mathscr{C}$ contains Arc $A \longrightarrow B$, then substituting it by $A \longleftarrow L \longrightarrow B$ results in a new path $\mathscr{C}^*$ of $\mathscr{G}^*$. As above, all the nodes except A, B, L have the same status. In addition, L cannot block path $\mathscr{C}^*$ and the connection at A is non-convergent in both $\mathscr{C}$ and $\mathscr{C}^*$ (because it has at most one parent, L). If the connection at B is convergent in $\mathscr{C}$, it is also convergent in $\mathscr{C}^*$, with its parent A substituted by L. If the connection at B in $\mathscr{C}$ is non-convergent, this means that it is of the form $A \longrightarrow B \longrightarrow C$, and in $\mathscr{C}^*$, its connection is $L \longrightarrow B \longrightarrow C$, also a non-convergent connection. Hence, overall, every path of $\mathscr{G}$ can be mapped into a path of $\mathscr{G}^*$ with the same status.

So, for all $U \in \mathbf{U}$ and $V \in \mathbf{V}$, we have that $\langle U \perp_{\mathscr{G}} V | \mathbf{W} \rangle \Longleftrightarrow \langle U \perp_{\mathscr{G}^*} V | \mathbf{W} \rangle$. ■

Proposition 4.3 *Let $\mathscr{B}^* = (\mathscr{G}^*, \Theta)$ be a Bayesian network such that $\mathscr{G}^* = (\mathbf{X}, \mathscr{E})$ and Arc $A \longrightarrow B$ belongs to $\mathscr{E}$. Let $L \in \mathbf{X}_H$ be such that $\mathbf{Ch}_{\mathscr{G}^*}(L) = \{A, B\} \subseteq \mathbf{X}_O$. Finally, let $\mathscr{G}$ be the graph resulting from the removal of L (and its outgoing arcs) from $\mathscr{G}^*$ and the addition of the set of arcs $\{X \longrightarrow B : X \in \mathbf{Pa}_{\mathscr{G}^*}(A) \setminus \{L\}\}$ (if $\mathscr{G}^*$ does not already contain them). Then, for any triple of disjoint subsets of variables $\mathbf{U}, \mathbf{V}, \mathbf{W}$ of $\mathbf{X}_O$, we have that:*

$$\langle \mathbf{U} \perp_{\mathscr{G}} \mathbf{V} | \mathbf{W} \rangle \Longleftrightarrow \langle \mathbf{U} \perp_{\mathscr{G}^*} \mathbf{V} | \mathbf{W} \rangle. \tag{II.2}$$

Proof of Proposition 4.3. As for Proposition 2, we just need to prove that, for any $U, V \in \mathbf{X}_O$, $U \neq V$, $\langle U \perp_{\mathscr{G}} V | \mathbf{W} \rangle \Longleftrightarrow \langle U \perp_{\mathscr{G}^*} V | \mathbf{W} \rangle$.

A simple path $\mathscr{C}^* = \langle X_1 = U, \ldots, X_k = V \rangle$ between U and V is active in $\mathscr{G}^*$ if and only if: i) all the nodes T with a convergent connection, either T or some of its descendants are in $\mathbf{W}$; and ii) nodes T with non-convergent connections do not belong to $\mathbf{W}$. Note that, by definition of $\mathscr{G}$, if $\mathbf{Desc}_{\mathscr{G}}(X)$ denotes the set of descendants of Node X in $\mathscr{G}$, then $\mathbf{Desc}_{\mathscr{G}}(X) = \mathbf{Desc}_{\mathscr{G}^*}(X)$ for any $X \in \mathbf{X} \setminus \{L\}$. Let us first show that, if there exists an active path in $\mathscr{G}^*$, then there also exists an active path in $\mathscr{G}$.

Let $\mathscr{C}^*$ be an active path of $\mathscr{G}^*$ that does not contain L, then it also exists in $\mathscr{G}$ and, since $\mathbf{Desc}_{\mathscr{G}}(X) = \mathbf{Desc}_{\mathscr{G}^*}(X)$ for all $X \in \mathbf{X} \setminus \{L\}$, it is also an active path of $\mathscr{G}$.

Assume now that path $\mathscr{C}$ contains L. Node L is equal neither to X_1 nor to X_k since U and V belong to $\mathbf{X}_O$. So, let $i \neq 1, k$, be the index such that, in path $\mathscr{C}^*$, $X_i = L$. By definition of $\mathscr{G}^*$, it holds that $\{X_{i-1}, X_{i+1}\} = \{A, B\}$. Without loss of generality, assume below that $X_{i-1} = A$ and $X_{i+1} = B$ (if this is the converse, consider the reversed path $\langle Y_1 = V, \ldots, Y_k = U \rangle$, which has the same d-separation status as $\mathscr{C}^*$, i.e., it is an active path).

If $X_1 = A$, then A is the first node of path $\mathscr{C}^*$. Let $\mathscr{C} = \langle X_1 = A, X_3 = B, X_4, \ldots, X_k \rangle$ then $\mathscr{C}$ belongs both to $\mathscr{G}^*$ and $\mathscr{G}$. In addition, the types of connection (convergent/non-convergent) of all the nodes $X_3, \ldots, X_{k-1}$ are the same in $\mathscr{C}^*$ and $\mathscr{C}$, and the sets of

descendants of these nodes are the same in $\mathcal{G}^*$ and $\mathcal{G}$. Finally, A and X_k cannot belong to $\mathbf{W}$ since they belong to $\mathbf{U}$ and $\mathbf{V}$ respectively. Hence, $\mathscr{C}$ is an active path in $\mathcal{G}$.

Suppose now that $X_1 \neq A$. Then, path $\mathscr{C}^*$ contains Node X_{i-2}, i.e., it contains Subsequences (X_{i-2}, A, L, B). If X_{i-2} is a child of A, then path $\mathscr{C} = (X_1, \ldots, X_{i-2}, A, B, X_{i+1}, \ldots, X_k)$ belongs to both $\mathcal{G}^*$ and $\mathcal{G}$. In addition, all the nodes in $\mathscr{C}$ have the same type of connection (convergent/non-convergent) as in $\mathscr{C}^*$. Therefore, since $\mathbf{D}\text{esc}_{\mathcal{G}}(X) = \mathbf{D}\text{esc}_{\mathcal{G}^*}(X)$ for all the nodes $X \in \mathscr{C}$, if $\mathscr{C}^*$ is an active path of $\mathcal{G}^*$, then $\mathscr{C}$ is an active path of $\mathcal{G}$.

Assume now that X_{i-2} is a parent of A. Let $\mathscr{C} = (X_1, \ldots, X_{i-2}, X_{i+1} = B, \ldots, X_k)$. $\mathscr{C}$ is a path in $\mathcal{G}$, and all of its nodes have the same type of connection (convergent/non-convergent) in $\mathcal{G}$ as their corresponding node of $\mathscr{C}^*$ in $\mathcal{G}^*$. Therefore, since $\mathbf{D}\text{esc}_{\mathcal{G}}(X) = \mathbf{D}\text{esc}_{\mathcal{G}^*}(X)$ for all the nodes $X \in \mathscr{C}$, $\mathscr{C}$ is an active path of $\mathcal{G}$.

To complete the proof, let us now show that if there exists an active path $\mathscr{C} = (X_1 = U, \ldots, X_k = V)$ in $\mathcal{G}$, then there also exists an active path $\mathscr{C}^*$ in $\mathcal{G}^*$.

If $B \notin \mathscr{C}$ then $\mathscr{C}$ can also belong to $\mathcal{G}^*$ because the only arcs that belong to $\mathcal{G}$ but not to $\mathcal{G}^*$ are Arcs $(Y \longrightarrow B)$ with $Y \in \mathbf{P}\text{a}_{\mathcal{G}^*}(A)$. As $\mathbf{D}\text{esc}_{\mathcal{G}}(X) = \mathbf{D}\text{esc}_{\mathcal{G}^*}(X)$ for all the nodes $X \in \mathscr{C}$ so $\mathscr{C}$ is also active in $\mathcal{G}^*$. Assume now that $B \in \mathscr{C}$. For the same reason as above, if none of the neighbors of B in $\mathscr{C}$ belong to $\mathbf{P}\text{a}_{\mathcal{G}^*}(A)$, then $\mathscr{C}$ is a path of $\mathcal{G}^*$ and is also active.

Let i be the index such that $X_i = B$. If some neighbors of B belong to $\mathbf{P}\text{a}_{\mathcal{G}}(A)$, two cases can be obtained: Case 1) exactly one neighbor of B belongs to $\mathbf{P}\text{a}_{\mathcal{G}}(A)$ and Case 2) exactly two neighbors of B belong to $\mathbf{P}\text{a}_{\mathcal{G}}(A)$. As for Case 1), without loss of generality, assume that $X_{i-1} \in \mathbf{P}\text{a}_{\mathcal{G}}(A)$ (else, reverse path $\mathscr{C}$). If $A \notin \mathbf{W}$, then path $\mathscr{C}^* = \langle X_1, \ldots, X_{i-1}, A, X_i = B, \ldots, X_k \rangle$ belongs to $\mathcal{G}^*$. In addition, the type of connection (convergent/non-convergent) of all the nodes in $\mathscr{C}^*$ except A and L are the same as their counterpart in $\mathscr{C}$. The connection at A is convergent and $A \in \mathbf{W}$, so A does not block $\mathscr{C}^*$ in $\mathcal{G}^*$. The connection at L is non-convergent and $L \notin \mathbf{W}$ since $L \notin \mathbf{X}_O$. So L does not block $\mathscr{C}^*$. Consequently, $\mathscr{C}^*$ is active in $\mathcal{G}^*$.

Consider now Case 2), i.e., the two neighbors of B in $\mathscr{C}$ belong to $\mathbf{P}\text{a}_{\mathcal{G}}(A)$. This means that X_{i-1}, B, X_{i+1} form a convergent connection. Since $\mathscr{C}$ is active, either B or some of its descendants belong to $\mathbf{W}$. But B is a child of A in both $\mathcal{G}$ and $\mathcal{G}^*$. Therefore, either A or some of its descendants belong to $\mathbf{W}$. As a consequence, if $\mathscr{C}^*$ is the path obtained from $\mathscr{C}$ by substituting B by A, then the connection at A in $\mathscr{C}^*$ is convergent and A does not block $\mathscr{C}^*$. For all the other nodes, the connections are similar in $\mathbf{C}$ and $\mathbf{C}^*$. Hence, $\mathbf{C}^*$ is active in $\mathcal{G}^*$. This completes the proof.

■

These two propositions address fundamental challenges in structure learning as shown in Figure II.4: distinguishing between graphs $\mathcal{G}^*$ that incorporate a latent variable L influencing two observable variables A and B (Figures II.4a, II.4c), and alternative structures $\mathcal{G}$ where L is absent but an arc directly connects A and B (Figure

II.4b), and an arc goes from C to B (Figure II.4d). Specifically, the proposition demonstrates that when L is the sole parent of A (or B), or when there exists an arc between the children of the latent variable, the resultant conditional independencies in the observable data do not provide a basis to discern whether the observed association between A and B is mediated by L or is a direct interaction in the absence of L.

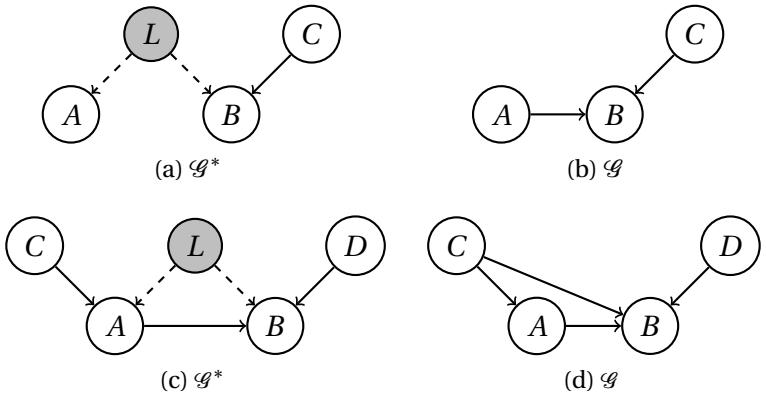


Figure II.4.: Indistinguishable Structures

Proposition 4.2, and 4.3 justify these configurations by showing that in the absence of additional parents for A and B, or a direct connection between A and B, structure learning algorithms cannot reliably infer the presence or absence of latent variables based solely on the observable data. By ensuring that children of latent variables in the network have specific structural signatures, we can provide a framework within which the existence and influence of latent variables can be more effectively inferred from the observable interactions, thereby avoiding the indistinguishability issue.

In summary, they highlight a critical aspect of causal inference in the presence of latent variables: the structural configuration of the model significantly impacts the identifiability of latent variables. By adopting a proper model, the proposed approach seeks to enhance the reliability of latent variable detection, thereby contributing to more accurate and interpretable causal structure learning from observational data.

Despite these interesting explanations of the basis of identifying latent variables within the learned structure, an important challenge remains: distinguishing original triangular relationships from those induced by latent variables. This distinction is crucial, as it directly influences our interpretation of the underlying causal mechanisms. There are some triangles in the learned structure that reflect direct interactions among the observable variables, whereas triangles created due to latent variables suggest hidden influences that need to be accounted for. In the forthcoming section, we discuss the methodologies and criteria employed to discern these two types of triangles. This analysis is crucial for refining our causal inference process, ensuring that the learned structures not only capture the observable data's intricacies but also accurately reflect the true causal relationships, including those obscured by latent

variables. This step marks a significant advancement in our ability to derive meaningful and reliable insights from complex data, further enhancing the interpretability and applicability of the learned causal models.

4.1.1. Genuine vs. Latent Triangles

The task of distinguishing between triangles that represent the direct relationships among the variables and latent-induced triangles within a DAG is a key point in the refinement of causal models derived from score-based learning algorithms. Our approach to this challenge is structured, beginning with the initial learning phase, where a primary structure is learned using a score-based algorithm. This foundational structure serves as a preliminary map, guiding us through the network of relationships among the observable variables.

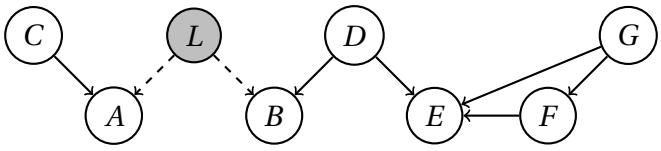
Subsequent to this initial phase, our focus shifts to the critical examination of triangular formations within the graph. This analysis aims to discern the nature of these triangles—whether they stand as genuine manifestations of direct interactions among variables or are mere artifacts of underlying latent variables. In instances where latent variables are inferred to be the cause of a triangle, strategic modifications are employed to realign the graph more closely with the original structure, similar to the transformation from configurations depicted in Figures II.3d or II.3e back to the more principal form represented in Figure II.2.

The final point of this process is the conversion of the resulting DAG into a CPDAG, a graph that encapsulates the maximal causality information permissible within the confines of the learned data. The forthcoming Section 4.2 is dedicated to an in-depth exploration of these steps. However, the central point of our current discussion is the methodology employed to accurately discriminate between the triangles, a task that depends on distinguishing *genuine* triangles—those inherent to the original network $\mathcal{G}^*$—from those induced by the presence of latent variables.

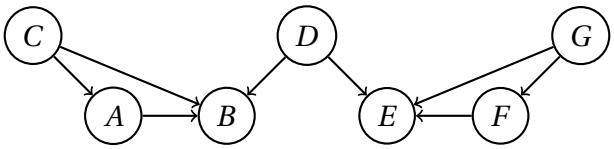
Definition 4.1 (Genuine Triangle) *Within the context of a DAG $\mathcal{G}^*$, over the variable set $\mathbf{X} = \mathbf{X}_O \cup \mathbf{X}_H$, a* genuine triangle *is defined as a clique $\langle A, B, C\rangle$ (as per Definition 1.4) where each vertex $\{A, B, C\}$ is an observable variable contained within $\mathbf{X}_O$.*

This classification underscores the genuine triangle as a direct reflection of the observable interactions, devoid of latent variable influences. To illustrate, consider Figure II.5 where the triangle $\langle E, F, G\rangle$ represents a genuine interaction within the graph. This scenario emphasizes a fundamental principle in causal inference: if $\mathcal{G}^*$ represents a perfect map of the underlying causal structure, then for any pair of variables within a genuine triangle, no subset of observable variables exists that can render these variables conditionally independent—thereby maintaining the integrity of their direct causal connection. In other words, for every pair (X, Y) of variables of the genuine triangles, there exists no d-separating set $\mathbf{Z} \subseteq \mathbf{X}_O \backslash \{X, Y\}$ that makes X and Y d-separated in $\mathcal{G}^*$ ($\langle X \perp_{\mathcal{G}^*} Y | \mathbf{Z}\rangle$) because $\mathcal{G}^*$ contains an arc between X and Y. So,

if dataset $\mathscr{D}$ is sufficiently large, we should not find any set $\mathbf{Z} \subseteq \mathbf{X}_O \backslash \{X, Y\}$ such that $X \perp\!\!\!\perp_{P^*} Y | \mathbf{Z}$.



(a) Original BN ($\mathscr{G}^*$)



(b) Possible Learned Structure ($\mathscr{G}$)

Figure II.5.: Genuine vs. Latent Triangles

The dataset $\mathscr{D}$, a projection of the more comprehensive $\mathscr{D}^*$ onto the space of observable variables $\mathbf{X}_O$, is instrumental in this analysis. It means $\mathscr{D}$ is generated by a distribution P over $\mathbf{X}_O$ defined as the marginal of P^* over $\mathbf{X}_O$, i.e., $P = \sum_{\mathbf{X}_H} P^*$. The equivalence of joint probabilities between $P(X, Y, \mathbf{Z})$ in $\mathscr{D}$ and $P^*(X, Y, \mathbf{Z})$ in $\mathscr{D}^*$ ensures that observable conditional independencies in $\mathscr{D}$ faithfully reflect those in the original dataset $\mathscr{D}^*$ ($X \perp\!\!\!\perp_P Y | \mathbf{Z}$ is equivalent to $X \perp\!\!\!\perp_{P^*} Y | \mathbf{Z}$). Consequently, within the context of genuine triangles, the dataset $\mathscr{D}$ is unlikely to reveal any conditional independencies among the triangle's vertices, thereby preserving the observable manifestations of their direct causal relationships. It indicates that it is not possible to find any pair of variables (X, Y) in the triangle such that there exists a set $\mathbf{Z} \subseteq \mathbf{X}_O \backslash \{X, Y\}$ and $X \perp\!\!\!\perp_{P^*} Y | \mathbf{Z}$.

Transitioning to the identification of latent triangles, we encounter pairs of nodes, such as (A, C) and (A, B) in Figure II.5b, that remain invariably connected, can not be d-separated by observable variables in $\mathscr{G}^*$. Conversely, for the pair (B, C) of Figure II.5b, there exist d-separating sets $\mathbf{Z} \subseteq \mathbf{X}_O \backslash \{B, C\}$ in $\mathscr{G}^*$ because $\mathscr{G}^*$ does not contain any arc between B and C. Note that $A \notin \mathbf{Z}$ otherwise path $\langle B, L, A, C \rangle$ would be active and $\langle B \not\perp\!\!\!\perp_{\mathscr{G}^*} C | \mathbf{Z} \rangle$. So, in terms of independences observable in dataset $\mathscr{D}$, there exist $\mathbf{Z} \subseteq \mathbf{X}_O \backslash \{A, B, C\}$ such $B \perp\!\!\!\perp_P C | \mathbf{Z}$ and, moreover, $B \not\perp\!\!\!\perp_P C | \mathbf{Z} \cup \{A\}$. This is the precise property we will use for discriminating between latent and genuine triangles:

Definition 4.2 (Latent Triangle) *Let $\mathscr{G}^*$ be a DAG over the set of variables $\mathbf{X} = \mathbf{X}_O \cup \mathbf{X}_H$ that generates the dataset $\mathscr{D}^*$. $\mathscr{D}$ is the projection of the $\mathscr{D}^*$ over $\mathbf{X}_O$, and $\mathscr{G}$ is a DAG which maximizes the BIC score over $\mathscr{D}$. A* latent triangle *is a clique $\langle A, B, C \rangle$ in $\mathscr{G}$ where it has* exactly one pair *of nodes that are independent given some set $\mathbf{Z} \subseteq \mathbf{X}_O$ but are dependent given $\mathbf{Z}$ union the third node of the triangle.*

This discernment between genuine and latent triangles, as described in Definition 4.2, is essential. It equips us with a robust framework for interpreting the learned

causal structures, as demonstrated in Figure II.5b with its depiction of both latent and genuine triangles. The forthcoming discussions will delve deeper into the methodologies employed for this critical discrimination, further enhancing the precision and reliability of causal inference from observational data.

4.2. The Algorithm

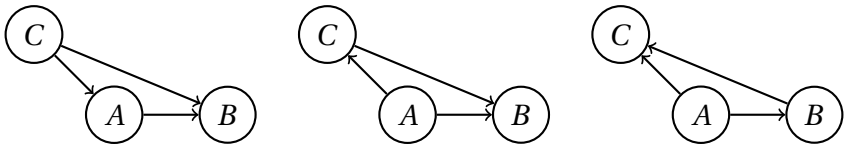
Based on the foundational concepts explained in the preceding sections, we can describe the procedure of our proposed method, starting with the initial learning of the primary structure through a score-based algorithm. This step takes advantage of the Bayesian Information Criterion (BIC) score to infer a preliminary graphical model from the data, setting the stage for the subsequent identification and analysis of latent structures within the learned graph. This procedure will be illustrated in the following sections: 4.2.1 Triangles Discrimination, 4.2.2 Finding d-Separating Sets, and 4.2.3 Modifying the Structure.

4.2.1. Triangles Discrimination

A key aspect of refining the initially learned structure involves the meticulous discrimination of triangles, identifying those that potentially include spurious dependencies indicative of latent variables. For instance, an arc such as $C \longrightarrow B$, as depicted in Figure II.5b, may suggest an underlying confounder. The challenge, therefore, lies in accurately locating these latent elements and their children, a task central to finding the true causal mechanisms within the network.

It is essential to recognize that the addition of the arc $C \longrightarrow B$ by score-based algorithms is based on the probabilistic dependence of B and C, given some set $\mathbf{Z} \supseteq \{A\}$. This dependence emerges due to the influence of a latent variable on its child A, thereby necessitating the inclusion of the arc $C \longrightarrow B$ to reflect this latent relationship within the learned structure. Identifying such an arc as indicative of a latent influence allows us to infer that the third node of the triangle, in this case, is a child of the latent confounder.

The determination of the second child of the confounder presents a more complex scenario. Consider the triangle configurations depicted in Figure II.6, each encoding equivalent d-separation properties and representing analogous sets of independencies. Theoretically, a score-based learning approach could learn any of these configurations, emphasizing the challenges involved in recognizing the true structure of the latent influences.



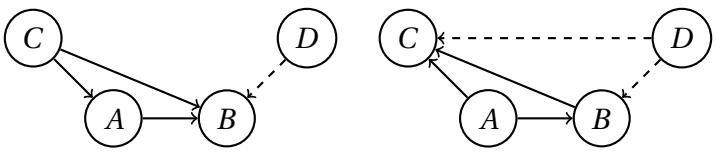
(a) Type 1 (b) Type 2 (c) Type 3

Figure II.6.: Possible Learned Latent Triangles

In the context of Figure II.6, the independent node pair mentioned in Definition 4.2 consistently involves B and C across all triangle types. Notably, triangles of Types 2 and 3 are characterized by the reversal of the arc $A \longrightarrow C$ as observed in $\mathcal{G}^*$. In the case of C possessing parents within $\mathcal{G}^*$, these relationships are inverted in $\mathcal{G}$, transforming parents into children to prevent the formation of new v-structures. Consequently, when encountering triangles of Types 2 or 3, C should have minimal parental connections, in contrast to B. This heuristic guides the orientation of the arc between B and C towards $B \longrightarrow C$, minimizing the BIC score's penalization. This principle was supported by empirical analysis, as detailed in Section 4.3, revealing that Type 2 triangles only allowed us to determine less than 0.5% of the latent confounders. Hence, to optimize the speed of our algorithm without decreasing its effectiveness, we have elected to focus on the discernment of Type 1 and Type 3 latent triangles. Rule 1 encapsulates the defining characteristics of these triangle configurations:

Rule 1 *Triangles of Type 1 (resp. Type 3) are such that* $C \overset{\frown}{\longrightarrow} A \longrightarrow B$ *(resp.* $A \overset{\frown}{\longrightarrow} B \longrightarrow C$*). For both types, there exists some set* $\mathbf{Z} \subseteq \mathbf{X}_O \backslash \{A, B, C\}$ *such that* $B \perp\!\!\!\perp_P C | \mathbf{Z}$ *and* $B \not\perp\!\!\!\perp_P C | \mathbf{Z} \cup \{A\}$*; and there exists no* $\mathbf{Z} \subseteq \mathbf{X}_O \backslash \{A, B, C\}$ *such that* $A \perp\!\!\!\perp_P B | \mathbf{Z}$ *or* $A \perp\!\!\!\perp_P C | \mathbf{Z}$*.*

In an ideal scenario where score-based algorithms flawlessly learn the underlying structure, the criteria outlined in Definition 4.2 would suffice for identifying latent triangles and localizing latent confounders. Nonetheless, practical applications of these algorithms are susceptible to errors, potentially leading to the erroneous detection of spurious latent triangles. To mitigate this vulnerability and strengthen the robustness of latent triangle identification, our approach extends beyond the examination of dependencies among the nodes A, B, and C, incorporating the analysis of interactions involving node D, the parent of B in $\mathcal{G}^*$. This careful consideration of node dependencies enhances the precision of our algorithm, ensuring a more reliable reconstruction of the latent structure.



(a) Type 1 (b) Type 3

Figure II.7.: Possible Learned Arcs Connected to the Latent Triangles

The patterns observed in Figure II.7, particularly the dotted arcs connecting node D with nodes A, B, and C, highlight the important roles these connections play in the structural learning process employed by score-based algorithms. In Type 1 triangles, the presence of an arc $D \longrightarrow B$ is anticipated, signifying that node B is likely to have a minimum of three parents within the learned graph $\mathcal{G}$. Consequently, Type 1 triangles are assigned as latent only when they satisfy this structural criterion in conjunction with the conditions outlined in Rule 1.

Algorithm 2: Learning with confounders.

Input: Dataset $\mathcal{D}$
Output: The CPDAG of the learnt CBN

$\mathcal{G} \leftarrow$ DAG learnt from $\mathcal{D}$ by a score-based algorithm
$\mathbf{T} \leftarrow$ the triangles of $\mathcal{G}$ of types 1 and 3 (see Rule 1)
// Get the latent triangles
$\mathbf{T}_1 \leftarrow \emptyset$ // latent triangles of type 1
$\mathbf{T}_3 \leftarrow \emptyset$ // latent triangles of type 3
foreach *triangle* $T = (A, B, C)$ *in* $\mathbf{T}$ **do**
    if T *is of Type 1 and* $|\boldsymbol{Pa}_{\mathcal{G}}(B)| >= 3$ **then**
        $\mathbf{T}_1 \leftarrow \mathbf{T}_1 \cup T$
    else if T *is of Type 3 and there exists* $D \in (\boldsymbol{Pa}_{\mathcal{G}}(B) \setminus \{A\}) \cup \boldsymbol{Ch}_{\mathcal{G}}(B) \setminus \{C\}$ *such that there exists* $\mathbf{Z} \subseteq \mathbf{X}_O \setminus \{C, D\}$ *such that* $A \in \mathbf{Z}$ *and* $D \perp\!\!\!\perp_P C | \mathbf{Z}$ **then**
        $\mathbf{T}_3 \leftarrow \mathbf{T}_3 \cup T$
// Recreate the latent variables
foreach *triangle* $T = (A, B, C)$ *in* $\mathbf{T}_1 \cup \mathbf{T}_3$ **do**
    Add a new node L (confounder) to $\mathcal{G}$
    Remove from $\mathcal{G}$ arc $A \rightarrow B$
    Remove from $\mathcal{G}$ the arc between B and C
    Add arcs $L \rightarrow A$ and $L \rightarrow B$ to $\mathcal{G}$
$\mathcal{M} \leftarrow$ the CPDAG of $\mathcal{G}$
return *CPDAG* $\mathcal{M}$

The analysis of Type 3 triangles indicates a more complex scenario. Beyond the expected arc $D \longrightarrow B$, an additional connection between D and C becomes requisite to accommodate the conditional dependency $\langle D \not\perp\!\!\!\perp_{\mathcal{G}^*} C | \{A, B\} \rangle$. This requirement gives rise to a new triangular formation (D, B, C), which, similar to the triangles of Types 1, 2, and 3, is subject to the same d-separation principles, thereby allowing for the potential reversal of arc orientations, preventing the introduction of directed cycles. As a result, in the learned structure $\mathcal{G}$, node D might emerge as a child of B, rather than its parent. Regardless of these configurations, a fundamental criterion persists: node D must exhibit independence from C given some set $\mathbf{Z} \subseteq \mathbf{X}_O$, which includes node A. Satisfying this criterion, alongside compliance with Rule 1, is crucial for the classification of Type 3 triangles as latent. This analytical framework is employed in

the formulation of Algorithm 2, containing the systematic approach to identifying latent structures within the learned graph.

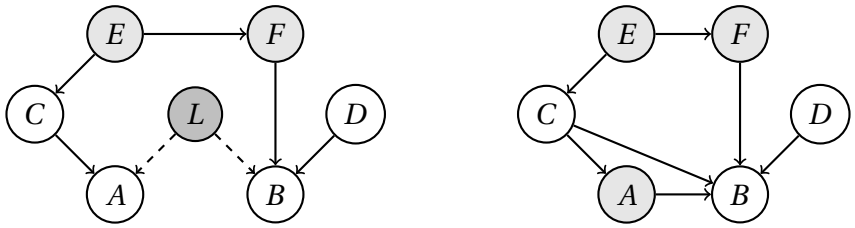
In Algorithm 2, Lines 6-7 is allocated for Type 1 triangles. In this part, the node B in each Type 1 triangle $T = (A, B, C)$ is being checked for a minimum of three parental connections, which is a necessary structural criterion for these triangles to be classified as latent. This process anticipates an arc from D to B, indicative of latent influences. Lines 8-9, assigned to our algorithm's approach for verification of Type 3 triangles, assessing the conditional independence between nodes D (if exists) and C, given a set $\mathbf{Z}$ containing A. This mirrors the complex interplay between direct and indirect influences within the graph, as discussed previously, thereby informing the algorithm's decision-making in categorizing triangles as latent. These steps are essential for discriminating the latent and genuine triangles.

The mentioned lines will categorize the latent triangles in two different sets but prior to these steps of the algorithm, it is necessary to find all Type 1 and Type 3 triangles within the graph (Line 2). This initial phase is governed by Rule 1, which shows the crucial role of d-separating sets in the identification process. The following section will elaborate on the mechanism and significance of d-separating sets for a comprehensive understanding of their application in our algorithm.

4.2.2. Finding d-Separating Sets

Regarding the basic methodologies outlined in Rule 1 and Algorithm 2, we now turn our attention to the critical task of identifying d-separating sets, $\mathbf{Z}$, within the observable variable set $\mathbf{X}_O$. These sets are instrumental in establishing the conditional independence of node pairs, an essential aspect of our algorithmic strategy.

To illustrate this concept, consider the example in Figure II.8. In this figure, all nodes are observable except L, and we encounter a Type 1 triangle with an extra indirect path that suggests a connection between B and C. In graph $\mathcal{G}^*$, nodes E and F act as d-separating sets for B and C, effectively 'blocking' the indirect path $\langle C, E, F, B\rangle$ between B and C. As we can see, the path $\langle B, L, A, C\rangle$ will be active given node A. This is not the case in graph $\mathcal{G}$. In this structure, to block all the paths between nodes B and C, we need to add node A to the d-separating set. This difference between the d-separating sets is essential for our algorithm to accurately discriminate between different types of triangles, as mandated by Rule 1.



(a) $\mathcal{G}^*$ (b) $\mathcal{G}$

Figure II.8.: d-separating set $\{E, F\}$ blocking the path between the vertices of a triangle

As mentioned in Rule 1, the essence of d-separation here is to examine whether the observed dependence between B and C is due to a latent confounder or not. In graph $\mathcal{G}^*$ one possible d-separation set is $\mathbf{Z} = \{E, F\}$, and $\mathbf{Z} \subseteq \mathbf{X}_O \setminus \{A, B, C\}$. We see that $B \perp_{\mathcal{G}^*} C|\mathbf{Z}$ and $B \perp_{\mathcal{G}} C|\mathbf{Z} \cup \{A\}$ but if $B \perp\!\!\!\perp_P C|\mathbf{Z}$, and $B \not\perp\!\!\!\perp_P C|\mathbf{Z} \cup \{A\}$ then we conclude that the clique $\langle A, B, C\rangle$ is a latent triangle.

By identifying sets like $\mathbf{Z} = \{E, F\}$ that can block these indirect links,we solidify the foundation for distinguishing latent triangles from genuine causal structures within the DAG. This step is not just a procedural necessity; it embodies the rigorous analytical process required to unveil the true causal mechanisms underlying the observed data.

We suggest two distinct algorithms to facilitate the identification of the set $\mathbf{Z}$. The first approach takes advantage of the principle that d-separation within I-maps implies conditional independence. By conducting a d-separation analysis on the graph $\mathcal{G}$, utilizing methodologies such as those outlined by Zander and Liśkiewicz 2020, we can uncover the required d-separating sets. The validity of this approach is described by the following proposition, which asserts the asymptotic consistency of the BIC score in identifying minimal I-maps[1]:

Proposition 4.4 *Let $\mathcal{D}^*$ be a dataset generated by some distribution P^* over $\mathbf{X}$ for which there exists a perfect map $\mathcal{G}^* = (\mathbf{X}, \mathcal{E})$, and let $\mathcal{D}$ be the projection of $\mathcal{D}^*$ over $\mathbf{X}_O$. Then, as $|\mathcal{D}| \longrightarrow \infty$, every DAG $\mathcal{G}$ maximizing the BIC score over $\mathcal{D}$ is a minimal I-map.*

Proof of Proposition 4.4. Lemma 4.1 considers pairs of nodes (A, B) such that there exists no directed path from B to A. So, A is not a descendant of B. In addition, in this lemma, there exists no arc between A and B, so A is not a parent of B. So Lemma 4.1 states that, in $\mathcal{G}$, every node is independent of its non-descendants given its parents.

Let $\mathcal{G}$ be a DAG maximizing the BIC score. Then it contains no arc $X \longrightarrow Y$ such that the graph $\mathcal{G}'$ resulting from the removal of $X \longrightarrow Y$ from $\mathcal{G}$ can maximize the BIC

[1]The asymptotic consistency of the BIC score was already known in other cases [Remco R Bouckaert 1993; Koller and Friedman 2009].

score. Indeed, as $|\mathscr{D}| \longrightarrow \infty$:

$$S(Y|\mathbf{Pa}_{\mathscr{G}}(Y)) \longrightarrow |\mathscr{D}| \times [\mathbf{I}_{P^*}(Y;\mathbf{Pa}_{\mathscr{G}'}(Y) \cup \{X\}) - \mathbf{H}_{P^*}(Y)] - \frac{\log|\mathscr{D}|}{2} dim(B|\mathbf{Pa}_{\mathscr{G}'}(Y) \cup \{X\})$$

It cannot be the case that $\mathbf{I}_{P^*}(Y;\mathbf{Pa}_{\mathscr{G}'}(Y) \cup \{X\}) < \mathbf{I}_{P^*}(Y;\mathbf{Pa}_{\mathscr{G}'}(Y))$ else $\mathscr{G}$ would not maximize the BIC score (since the term in $\log|\mathscr{D}|$ is infinitely smaller than $|\mathscr{D}|$). But, for every probability distribution Q and every $U, V, \mathbf{Z}$, it always holds that $\mathbf{I}_Q(U;V,\mathbf{Z}) \geq \mathbf{I}_Q(U;\mathbf{Z})$, with equality only if $U \perp\!\!\!\perp_Q V|\mathbf{Z}$. So, necessarily, $\mathbf{I}_{P^*}(Y;\mathbf{Pa}_{\mathscr{G}'}(Y) \cup \{X\}) = \mathbf{I}_{P^*}(Y;\mathbf{Pa}_{\mathscr{G}'}(Y))$. As a consequence, we have that

$$\begin{aligned} S(Y|\mathbf{Pa}_{\mathscr{G}}(Y)) - S(Y|\mathbf{Pa}_{\mathscr{G}'}(Y)) &\approx \frac{\log|\mathscr{D}|}{2}[dim(B|\mathbf{Pa}_{\mathscr{G}'}(Y) \cup \{X\}) - dim(B|\mathbf{Pa}_{\mathscr{G}'}(Y))] \\ &\approx \frac{\log|\mathscr{D}|}{2}(|\Omega_X| - 1) \times (|\Omega_Y| - 1) \times (|\Omega_{\mathbf{Pa}_{\mathscr{G}'}(Y)}|) > 0. \end{aligned}$$

Hence $\mathscr{G}'$ cannot maximize the BIC score. So $\mathscr{G}$ is minimal. Overall, by Corollary 4, p. 120, of Pearl 1988, $\mathscr{G}$ is a minimal I-map. ■

However, when the database is not "too" large, and Graph $\mathscr{G}$ is not highly trustworthy, another option is to exploit Algorithm 3 which adds iteratively to $\mathbf{Z}$ the variable X that allows to reduce the most the dependence between the pair of nodes U, V until an independence is inferred. In essence, this approach is followed by MIIC [Verny, Sella, Affeldt, et al. 2017], estimating dependences through an information theoretic criterion.

Algorithm 3 outlines the procedure to ascertain the existence of a d-separating set $\mathbf{Z}$ that makes nodes U and V conditionally independent. The algorithm initiates by setting $\mathbf{Z}$ to $\mathbf{C}$, ensuring it encompasses all compulsory variables, and concurrently updates $\mathbf{F}$ to exclude specific variables from $\mathbf{Z}$. An early termination criterion is

Algorithm 3: Check whether there exists $\mathbf{Z}$ s.t. $U \perp\!\!\!\perp V|\mathbf{Z}$

Input: nodes U, V, Dataset $\mathscr{D}$, max size h of $\mathbf{Z}$, risk level α, sets $\mathbf{C}$ and $\mathbf{F}$ s.t. $\mathbf{Z} \supseteq \mathbf{C}$ and $\mathbf{Z} \cap \mathbf{F} = \emptyset$

Output: A Boolean indicating whether $\mathbf{Z}$ exists

$\mathbf{Z} \leftarrow \mathbf{C}$; $\mathbf{F} \leftarrow \mathbf{F} \cup \{U, V\}$
$\delta \leftarrow (|\Omega_U| - 1) \times (|\Omega_V| - 1) \prod_{X \in \mathbf{C}} |\Omega_X|$
if $|\mathbf{Z}| > h$ **then return** *False*;
if $G2(U, V|\mathbf{Z}) < \chi^2_\delta(\alpha)$ **then return** *True*;
while $|\mathbf{Z}| < h$ **do**
  $X = \text{argmin}\{G2(U, V|\mathbf{Z} \cup \{Y\}) : Y \in \mathbf{X}_O \setminus (\mathbf{Z} \cup \mathbf{F})\}$
  $\mathbf{Z} \leftarrow \mathbf{Z} \cup \{X\}$; $\delta \leftarrow \delta \times |\Omega_X|$
  if $G2(U, V|\mathbf{Z}) < \chi^2_\delta(\alpha)$ **then return** *True*;
return *False*

evaluated to check if the size of $\mathbf{Z}$ surpasses the predefined limit h or if G^2 statistic shows U and V are independent given $\mathbf{Z}$, using the chi-squared distribution threshold.

The core of the algorithm lies within an iterative loop that strategically augments $\mathbf{Z}$ with variables that minimally preserve the dependence between U and V, as measured by the G^2 statistic. This loop continues until either $\mathbf{Z}$ reaches its capacity at h or the addition of variables no longer contributes to establishing conditional independence between U and V.

The algorithm's termination results in a Boolean output, affirming the identification of a d-separating set $\mathbf{Z}$ if U and V become conditionally independent; otherwise, it denotes the absence of such a set. This process is performed based on the statistical measures and acts as the algorithm's utility in discerning conditional independencies within the dataset $\mathscr{D}$, thereby facilitating a better understanding of the underlying causal structures.

Here, we suggest to simply exploit the G2 formula of the conditional G-test $G2(U,V|\mathbf{Z})$, defined as:

$$G2(U,V|\mathbf{Z}) = 2\sum_{u,v,\mathbf{z}} N_{uv\mathbf{z}} \ln \frac{N_{uv\mathbf{z}} N_{\mathbf{z}}}{N_{u\mathbf{z}} N_{v\mathbf{z}}}, \tag{II.3}$$

where $N_{uv\mathbf{z}}$ is the number of occurrences of $U=u, V=v, \mathbf{Z}=\mathbf{z}$ in Database $\mathscr{D}$, $N_{u\mathbf{z}} = \sum_v N_{uv\mathbf{z}}$, $N_{v\mathbf{z}} = \sum_u N_{uv\mathbf{z}}$ and $N_{\mathbf{z}} = \sum_u N_{u\mathbf{z}}$.

In this context, the variable X that minimizes G2 is selected for inclusion in $\mathbf{Z}$. The relationship between G2 and the BIC score, described in the subsequent proposition, offers a precise basis for this selection process:

Proposition 4.5 *For every variables U, V and set of variables $\mathbf{Z}$, $G2(U,V|\mathbf{Z})$ is equal to:*

$$2 \times \left(S(U|V,\mathbf{Z}) - S(U|\mathbf{Z}) + \frac{1}{2}\log(|\mathscr{D}|)\delta \right), \tag{II.4}$$

where:
$\delta = (|\Omega_U|-1) \times (|\Omega_V|-1) \times |\Omega_{\mathbf{Z}}|$,
Ω_U *(resp.* Ω_V, $\Omega_{\mathbf{Z}}$*) is the domain of variable* U *(resp.* V, $\mathbf{Z}$*) and,*
$S(.|.)$ *is the BIC score.*

Proof of Proposition 4.5. Let $\Delta_{dim} = \frac{1}{2}\log(|\mathscr{D}|)[dim(U|V,\mathbf{Z}) - dim(U|\mathbf{Z})] = \frac{1}{2}\log(|\mathscr{D}|)\delta$, with $\delta = (|\Omega_U|-1) \times (|\Omega_V|-1) \times |\Omega_{\mathbf{Z}}|$. Then we have that:

$$\begin{aligned} S(U|V,\mathbf{Z}) - S(U|\mathbf{Z}) &= \sum_{u,v,\mathbf{z}} N_{uv\mathbf{z}} \left[\log\left(\frac{N_{uv\mathbf{z}}}{N_{v\mathbf{z}}}\right) - \log\left(\frac{N_{u\mathbf{z}}}{N_{\mathbf{z}}}\right) \right] - \Delta_{dim} \\ &= \sum_{u,v,\mathbf{z}} N_{uv\mathbf{z}} \log\left(\frac{N_{uv\mathbf{z}} N_{\mathbf{z}}}{N_{u\mathbf{z}} N_{v\mathbf{z}}}\right) - \Delta_{dim} \end{aligned}$$

So we have that:

$$2 \times (S(U|V,\mathbf{Z}) - S(U|\mathbf{Z}) + \Delta_{dim}) = 2 \sum_{u,v,\mathbf{z}} N_{uv\mathbf{z}} \log\left(\frac{N_{uv\mathbf{z}} N_{\mathbf{z}}}{N_{u\mathbf{z}} N_{v\mathbf{z}}}\right) = G2(U,V|\mathbf{Z}).$$

■

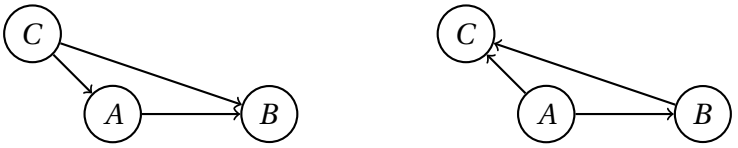
According to this proposition, if U and V are conditionally independent given $\mathbf{Z}$, the G2 statistic follows a χ^2 distribution with δ degrees of freedom. Given a predefined significance level α, typically set at 0.05, U and V are considered independent if the G2 value falls below the χ^2 distribution's critical threshold, χ^2_α.

4.2.3. Modifying the Structure

Having explained the discrimination between genuine and latent triangles and finding the precise placement of latent variables, we now introduce the structural modification within the learned graph, as encapsulated in the subsequent lines of Algorithm 2 (Lines 10 to 15). This phase is essential, focusing on the strategic alteration of the graph to accurately reflect the inferred latent variables and their connections.

Upon identifying Type 1 and Type 3 triangles, each latent triangle prompts adding a new node to the graph to represent the latent variable. Subsequently, we undertake the removal of arcs that do not correspond to the original causal structure; precisely, the arc between the children of the latent variable ($A \rightarrow B$), and the arc linking an observable parent of one child to the other child (B and C). As previously discussed, these arcs are artifacts introduced by the presence of latent variables, falsely implying direct causal links where none exist to satisfy the d-separation criteria among the triangle nodes. An illustrative depiction of these modification steps for both Type 1 and Type 3 triangles is provided in Figure II.9.

Found latent triangles:



(a) Type 1 (b) Type 3

Adding new node L (Line 14):



(c) Type 1 (d) Type 3

Remove arc $A \rightarrow B$ (Line 15):



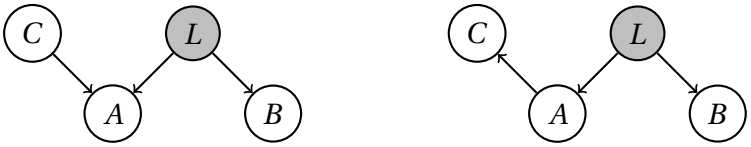
(e) Type 1 (f) Type 3

Remove the arc between B and C (Line 16):



(g) Type 1 (h) Type 3

Add arcs $L \rightarrow A$ and $L \rightarrow B$ (Line 17):



(i) Type 1 (j) Type 3

Figure II.9.: Graph Modification Steps of Algorithm 2

The final part of our algorithmic process is the transition of the adjusted graph $\mathcal{G}$ to a CPDAG (Line 15 of Algorithm 2). This transformation is crucial, as CPDAGs help us clearly see and understand how different factors cause each other in our graph. They show us the direction of these causal relationships and highlight areas where we might not be sure how things are connected. The next part will explain how we change the graph into a CPDAG. This will help us get a CPDAG that shows us the best picture of the cause-and-effect relationships we've found.

As discussed in Section 2.2.2 of Part I, CPDAGs are characterized by their definitive arc orientations, which are consistent across all I-equivalent graphs sharing the same

underlying structure. To transform a DAG into a CPDAG, we can apply Meek's rules [Meek 1995], which entail the following steps:

1. Begin with the skeleton of the DAG by converting all directed arcs into (undirected) edges and then orienting the v-structures (immoralities) present within the DAG. This initial step yields a partially oriented DAG (PDAG), which is not yet complete.
2. To prevent the formation of new v-structures, when there exist some triples like X—$Y \longleftarrow Z$, orient it as $X \longleftarrow Y \longleftarrow Z$.
3. To maintain acyclicity, if there is a directed path from X to Y, the edge between them must be oriented as $X \longrightarrow Y$ to prevent the creation of cycles.
4. The final step involves a careful review to ensure that orienting an edge does not introduce new v-structures or cycles. If such a situation arises, the edge must be oriented in the opposite direction to avoid these issues.

In Figure II.10a, we show the output DAG after Line 14 of Algorithm 2. The following figures illustrate how we change this DAG into a CPDAG, step by step.

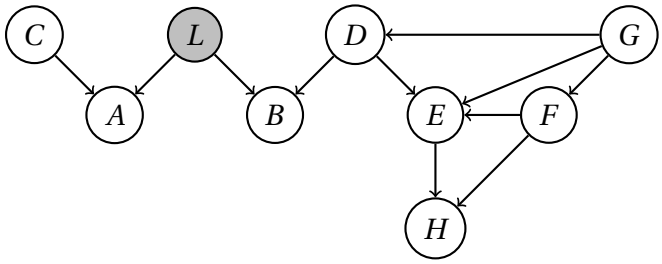
The red arrows in Figure II.10 highlight the changes we make at each step. We base all steps on earlier explanations. In Figures II.10c, II.10d, and II.10e, it is important to note that flipping the direction of a red arrow could cause problems like new v-structures or cycles, as noted next to the figures. Figure II.10e might need more detail. It looks like flipping an arrow does not cause issues, but if we add an arrow from E to G, then we have to orient the edges from D to G and F to G to prevent cycles. These changes could create a new v-structure ($D \longrightarrow G \longleftarrow F$), so we direct the edge from G to E to avoid new v-structures and cycles.

4.2.4. Complexity Analysis

In the pursuit of a comprehensive understanding of our proposed methodology, it is imperative to go through the computational complexity of Algorithm 2. This analysis not only evaluates the efficiency and scalability of the algorithm but also provides insights into its practical applicability to datasets of varying sizes and dimensions. The forthcoming discussion aims to dissect the algorithmic steps, evaluating their individual and collective computational demands, thereby presenting a better view of the algorithm's performance characteristics.

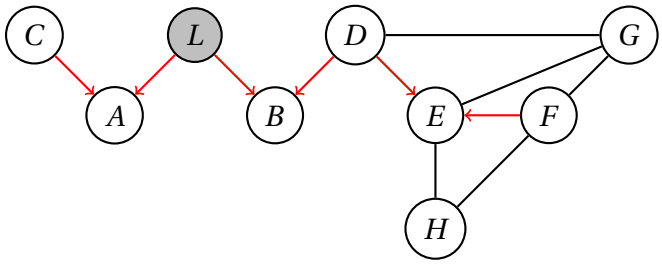
In Section 4.2.2, we introduced two methodologies for the identification of d-separating sets, essential for discerning conditional independencies within the graph. Given the adoption of Algorithm 3 for this critical task, the computational complexity of Algorithm 2 can be calculated by the following proposition. This proposition illustrates the algorithm's efficiency, thereby offering a comprehensive understanding of the algorithm's computational demands in various scenarios.

Modified Learned DAG:



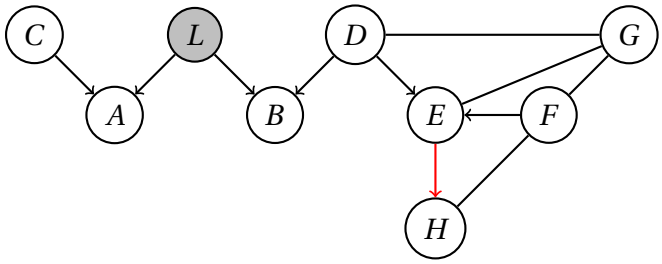
(a) $\mathcal{G}$

1. Orient the v-structures in the skeleton:



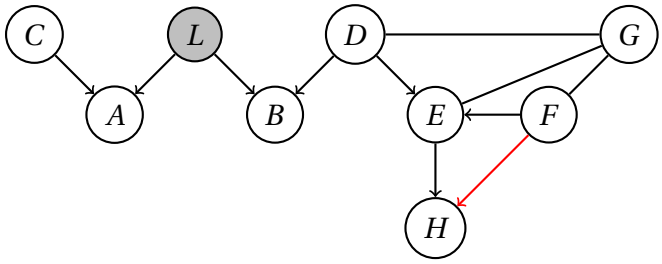
(b) $\mathcal{G}_1$

2. Avoid creating new v-structures:



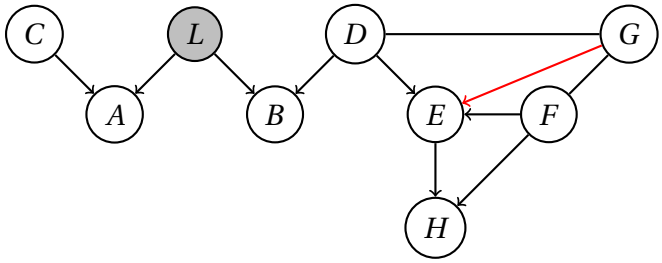
(c) $\mathcal{G}_2$

3. Avoid creating new cycle:



(d) $\mathcal{G}_3$

4. Avoid creating new v-structure and cycle:



(e) $\mathcal{G}_4$

Figure II.10.: Converting a DAG to a CPDAG

Proposition 4.6 *Assume that the existence of d-separating sets $\mathbf{Z}$ is checked with Algorithm 3 with h the maximal size allowed for $\mathbf{Z}$. Let n and m denote the number of nodes and arcs of $\mathcal{G}$, respectively. Let k be the maximum number of parents and children of the nodes in $\mathcal{G}$. Then the time complexity of Algorithm 2 over dataset $\mathcal{D}$ is $O(n^2k^3h|\mathcal{D}| + m\log n)$.*

Proof of Proposition 4.6. Given the structure and objectives of Algorithm 2, the complexity analysis depends on several key parameters: the number of nodes (n) and arcs (m) in the graph $\mathcal{G}$, the maximum number of parents and children (k) for any node, and the maximal size (h) for the d-separating sets $\mathbf{Z}$.

Algorithm 3, employed for conditional independence tests, iterates up to h steps, each evaluating the potential inclusion of a variable X to the set $\mathbf{Z}$. For each X, a conditional G-test is performed, leading to a complexity of $O(nh|\mathcal{D}|)$ for Algorithm 3.

Considering there are at most nk^2 triangles in $\mathcal{G}$, and evaluating each for latent characteristics (as per Rule 1) involves applying Algorithm 3 to all three arcs of a triangle, the execution time for this phase scales as $O(n^2k^2h|\mathcal{D}|)$.

On Line 2, the information about which pair of nodes is independent is cached. So, determining their types on Lines 6 and 8 is performed in $O(1)$ time. Determining whether $\mathbf{Pa}_{\mathcal{G}}(X_3) \geq 3$ can also be done in $O(1)$. As for Lines 8, there are at most k nodes Y to examine each using Algorithm 3. So, the time complexity of Loop 5–9 is $O(n^2k^3h|\mathcal{D}|)$. There are at most nk^2 triangles in $\mathcal{G}$ hence $O(nk^2)$ iterations of the *for* loop of Lines 10-14. Each instruction on lines 11 to 14 can be performed in $O(1)$ times; hence the loop of Lines 10-14 can be performed in $O(n^2k^3h|\mathcal{D}|)$.

The subsequent step, including the transformation of $\mathcal{G}$ into a CPDAG contribute additional complexity equals $O(m\log n)$ [David Maxwell Chickering 1995]. However, the dominant factor remains the d-separating set determination, resulting in an overall time complexity for Algorithm 2 of $O(n^2k^3h|\mathcal{D}| + m\log n)$. ■

Proposition 4.6 specifies the time complexity involved in executing Algorithm 2, taking into account the maximal allowable size for $\mathbf{Z}$, denoted by h, alongside the graph's structural parameters, including the number of nodes n, arcs m, and the maximum count of parents and children k for any node within $\mathcal{G}$. This comprehensive analysis ensures a robust and efficient approach to d-separating set determination, thereby enhancing the algorithm's overall efficacy in latent structure identification.

4.3. Experiments

This section rigorously outlines the experimental setup, datasets employed, and the metrics used to evaluate our proposed algorithm's performance. Through a series of experiments, we aim to not only validate the effectiveness and robustness of our method in uncovering the underlying causal structures but also to compare its ability

against established benchmarks in the field. The following discussion provides a comprehensive account of the experimental design, execution, and critical analysis of the results, offering insightful perspectives on the practical applicability and potential impact of our algorithm in advancing causal learning research.

Building upon the foundational setup described, we go through a series of investigations utilizing some classical benchmark Causal Bayesian Networks (CBNs) such as Child ($|\mathbf{X}_O| = 20$), Water ($|\mathbf{X}_O| = 32$), Insurance ($|\mathbf{X}_O| = 27$), Alarm ($|\mathbf{X}_O| = 37$), and Barley ($|\mathbf{X}_O| = 48$), sourced from the renowned BNLearn Bayesian network repository[2]. These experiments are designed to underscore the capability of our algorithm in accurately identifying latent confounders within diverse and complex network structures.

Network	# Nodes	# Arcs	# Parameters
Child	20	25	230
Alarm	37	46	509
Insurance	27	52	1008
Water	32	66	10083
Barley	48	84	114005

Table II.1.: Properties of the CBNs employed for the experiments

Each CBN is chosen to perform an accurate examination in our experimental setup. To this end, new latent variables, denoted as L_i, are strategically introduced into the selected CBN. These latent variables are designed to act as confounders within a semi-Markovian causal model framework. Specifically, for each L_i, a pair of nodes from the set $\mathbf{X}_O$ is randomly selected to assume the role of L_i's children, thereby inserting these latent variables into the causal structure.

To cohere to the theoretical foundations laid out in this study, the initial step of Algorithm 2, specifically the learning of the primary structure, is executed utilizing the CPBayes exact score-based learning algorithm, as detailed in Van Beek and Hoffmann 2015. This phase necessitates the transformation of Datasets $\mathscr{D}$ into what are termed as 'instances'. These instances contain all the possible sets of nodes that CPBayes will consider as potential parent sets, and to mitigate computational complexity, this requires limiting the number of possible parents of the nodes. In our empirical experiments, this threshold is set to four, a decision underpinned by two critical observations: firstly, the original CBNs—prior to the integration of confounders—did not feature any node with a parental count exceeding four. Secondly, this constraint significantly reduces the computational workload; further information is available in [Van Beek and Hoffmann 2015].

Limiting the number of possible parents of the nodes up to four has a necessary consequence: each child of the latent variable must have one or two existing parents

[2] https://www.bnlearn.com/bnrepository

within $\mathbf{X}_O$, ensuring compatibility with the algorithm's structural modifications. This criterion is essential as the score-based algorithm has the propensity to append two additional parents, as depicted in Figure II.3d, (A and C) are the new parents of the children (B) of L_i, Thus, this selective approach to choosing L_i's children, constrained by their parental count within $\mathbf{X}_O$, is instrumental in maintaining the feasibility of the experimental design.

In the process of finding out the relationships wherein nodes A and B emerge as children of a latent confounder L_i, the underlying principle utilized by various learning algorithms rely on the conditional dependence of A and B given any set $\mathbf{Z}$. This crucial aspect imposes certain constraints on formulating the conditional probability distributions (CPDs) for L_i and its children. The generation of these CPDs demands a distinguished approach, since a mere random allocation could accidentally hide the conditional dependencies intrinsic to the causal framework.

Consider a scenario where the CPDs of L_i, A, and B are uniformly randomly generated. In such a case, the entries within the joint probability distribution P encompassing A, B, and their parents would be generated from a uniform distribution. While apparently correct, this randomness could potentially distort the conditional dependencies, complicating the task of accurately identifying the latent structures. Consequently, a methodically structured approach to generating these distributions is needed to preserve the inherent conditional dependencies essential for the reliable detection of latent confounders.

Validation of the conditional (in)dependence between A and B given various subsets $\mathbf{Z}$, which are subsets of the parents of A and B, essentially entails marginalizing certain variables from P. This process, in essence, involves the aggregation of values within P, a mathematical operation that bears significant implications. The sum of two independently distributed uniform variables follows a triangular distribution, and it tends to a normal distribution with the addition of further variables, as shown by the central limit theorem. Consequently, the marginal distributions derived from P predominantly exhibit values clustered around the mode rather than the tails of the distribution. This clustering effect makes A and B appear independent, thereby hiding their true relationship as children of a latent confounder cannot be detected by any learning algorithms.

To mitigate this phenomenon, the CPDs of A, B, and L_i necessitate a distinct generation strategy. The CPD of L_i is set to a uniform distribution to maximize the chance of a dependent relationship between A and B. For A (resp. B), for each value of its parents, the CPD of A (resp. B) is set to a mixture of a Dirichlet distribution whose hyperparameters α_i are all set to 4 and a Dirac distribution. The weighting assigned to the Dirac distribution is chosen randomly within the range of 2/3 to 5/6, a strategic decision aimed at limiting the influence of the central limit theorem without entirely discarding its effects.

To validate the efficacy of this approach, an empirical assessment is conducted to ascertain if the mutual information between A and B, alongside their conditional mutual information given by their parents, surpasses predefined thresholds. These thresholds are not arbitrarily chosen but are derived from the average mutual and conditional mutual information values among node pairs with a shared parent in the original CBN, ensuring a grounded basis for comparison. Only when these conditions are met are the CPDs of A and B considered suitable for inclusion in our experimental evaluation, thereby ensuring a realistic and robust testing environment.

After adding latent variables to each CBN as described above, we need to produce data corresponding to the structures. To generate data for these enhanced CBNs, we employ the pyAgrum 1.8.3 library [Ducamp, Gonzales, and Wuillemin 2020], a robust tool for Bayesian network modeling. The generated dataset, denoted as $\mathscr{D}^*$, encompasses both observable and latent variables. However, for the purpose of learning algorithms, we focus solely on the observable portion, $\mathscr{D}$, which is essentially a subset of $\mathscr{D}^*$ where latent variables have been excluded.

Our evaluation involves a comparative analysis of three distinct learning algorithms: our proposed method (Algorithm 2, henceforth referred to as '**Algo 1**'), the MIIC algorithm [Verny, Sella, Affeldt, et al. 2017], and the well-established FCI algorithm [Peter Spirtes, Clark Glymour, Richard Scheines, et al. 1993]. For MIIC we utilize pyAgrum's implementation with an NML correction to enhance its precision. Meanwhile, for FCI, the implementation is sourced from the Python causal-learn 0.1.3.3 package [Y. Zheng, Huang, Chen, et al. 2023], known for its comprehensive causal analysis capabilities.

To ensure the robustness of our findings, we conduct each experiment across 50 distinct CBN/dataset combinations, averaging the results to obtain a reliable performance metric. These experiments are executed on a high-performance computing setup featuring an Intel Xeon Gold 5218 CPU and 128GB of RAM, ensuring that computational intensity does not our analysis.

In our evaluation framework, we utilize the terms 'ok' and '¬ok' to respectively denote the number of correctly identified confounders and the number of incorrectly identified confounders. Based on these, we define some key metrics to assess the performance of different methods on various datasets derived from different CBNs with added confounders:

- **Precision (prec.)**: This metric evaluates the accuracy of the identified confounders, calculated as the ratio of correctly identified confounders ('ok') to the total number of identified confounders ('ok' + '¬ok'):

$$\text{Precision} = \frac{\text{ok}}{\text{ok} + \neg\text{ok}}$$

- **Recall**: This metric assesses the algorithm's capability to identify all actual confounders, defined as the ratio of correctly identified confounders ('ok') to the

		Algo 1						**MIIC**					**FCI**				
CBN	$\|\mathscr{D}\|$	ok	¬ok	prec.	recall	F1	time	ok	¬ok	prec.	recall	F1	ok	¬ok	prec.	recall	F1
child	5000	0.50	0.04	**0.93**	0.25	0.39	0.02	1.40	0.74	0.65	0.70	**0.68**	1.76	3.88	0.31	**0.88**	0.46
	10000	0.76	0.20	**0.79**	0.38	0.51	0.04	1.59	0.69	0.70	0.80	**0.74**	1.86	2.92	0.39	**0.93**	0.55
	20000	1.10	0.12	**0.90**	0.55	**0.68**	0.07	1.38	0.83	0.62	0.69	0.65	1.74	2.12	0.45	**0.87**	0.59
	50000	1.42	0.18	**0.89**	0.71	**0.79**	0.14	1.45	0.71	0.67	0.72	0.70	1.74	1.08	0.62	**0.87**	0.72
water	5000	0.16	1.58	0.09	0.08	0.09	0.02	0.35	2.78	**0.11**	**0.17**	**0.14**	0.34	8.26	0.04	**0.17**	0.06
	10000	0.44	0.04	**0.92**	0.22	0.35	0.05	1.25	1.71	0.42	0.63	**0.50**	1.54	5.42	0.22	**0.77**	0.34
	20000	0.80	0.22	**0.78**	0.40	**0.53**	0.14	1.40	1.88	0.43	0.70	**0.53**	1.60	6.00	0.21	**0.80**	0.33
	50000	1.16	0.62	**0.65**	0.58	**0.61**	0.16	1.40	1.94	0.42	0.70	0.52	1.56	7.86	0.17	**0.78**	0.27
insu-rance	5000	0.20	0.14	**0.59**	0.10	0.17	0.06	1.65	4.05	0.29	0.83	**0.43**	1.88	8.88	0.17	**0.94**	0.29
	10000	0.30	0.26	**0.54**	0.15	0.23	0.11	1.59	3.57	0.31	0.80	**0.44**	1.96	8.66	0.18	**0.98**	0.31
	20000	0.60	0.30	**0.67**	0.30	0.41	0.24	1.81	4.17	0.30	0.90	**0.45**	1.94	8.46	0.19	**0.97**	0.31
	50000	1.16	0.52	**0.69**	0.58	**0.63**	0.53	1.85	3.91	0.32	0.92	0.48	1.90	8.72	0.18	**0.95**	0.30
alarm	5000	0.36	0.42	**0.46**	0.18	0.26	0.07	1.26	1.62	0.44	0.63	**0.52**	1.48	4.70	0.24	**0.74**	0.36
	10000	0.84	0.56	**0.60**	0.42	0.49	0.15	1.52	1.02	**0.60**	0.76	**0.67**	1.58	4.14	0.28	**0.79**	0.41
	20000	1.08	0.66	0.62	0.54	0.58	0.24	1.55	0.65	**0.70**	0.78	**0.74**	1.72	3.42	0.33	**0.86**	0.48
	50000	1.34	1.16	0.54	0.67	0.60	0.57	1.67	1.04	**0.62**	**0.84**	**0.71**	1.68	2.92	0.37	**0.84**	0.51
barley	5000	0.00	0.00	–	0.00	–	0.02	0.60	3.22	**0.16**	0.30	**0.21**	1.58	20.68	0.07	**0.79**	0.13
	10000	0.06	0.00	**1.00**	0.03	0.06	0.04	0.76	2.40	0.24	0.38	**0.29**	1.78	19.82	0.08	**0.89**	0.15
	20000	0.28	0.00	**1.00**	0.14	0.25	0.09	1.08	1.98	0.35	0.54	**0.43**	1.86	16.54	0.10	**0.93**	0.18
	50000	0.66	0.10	**0.87**	0.33	**0.48**	0.27	1.36	3.75	0.27	0.68	0.38	1.86	14.82	0.11	**0.93**	0.20

Table II.2.: Confounders found for different dataset sizes and CBNs with 2 Boolean confounders

total actual confounders ('L_i') added to the original CBN:

$$\text{Recall} = \frac{\text{ok}}{\#L_i}$$

- **F1 Score**: Harmonizes precision and recall, computed as the harmonic mean of the two. The F1 score is:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Table II.2 showcases a comparative analysis of the three algorithms' performance on datasets of varying sizes, where each original CBN has been augmented with two confounders of domain size 2. The table shows the number of correctly ('ok') and incorrectly ('¬ok') identified confounders, alongside the computed precision, recall, and F1 score for each scenario. Furthermore, the 'time' column indicates the average computation time (in seconds) necessitated by Algo 1 for the identification of confounders (specifically, Lines 2 to 15), which only marginally extends the overall structure learning computation duration.

The performance of FCI, MIIC, and Algo 1 in identifying latent confounders varies significantly across different metrics. FCI, with its comprehensive search strategy, demonstrates a high *recall*, indicating its proficiency in uncovering a majority of latent confounders (high ok values). However, this extensive identification approach leads to a lower *precision* due to the misclassification of numerous non-confounder variables

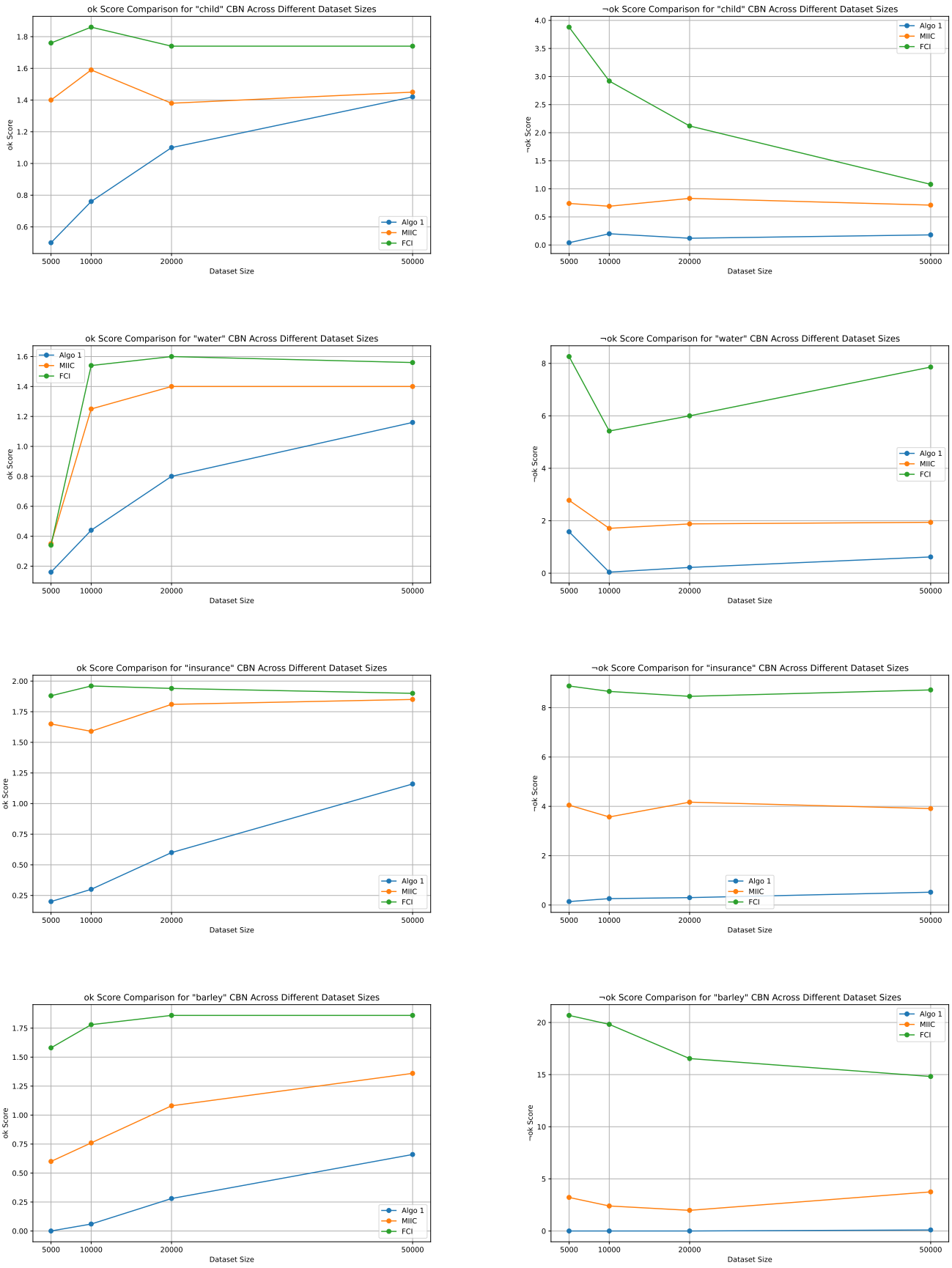


Figure II.11.: Comparative plots for "ok" and "¬ok" metric across different CBNs.

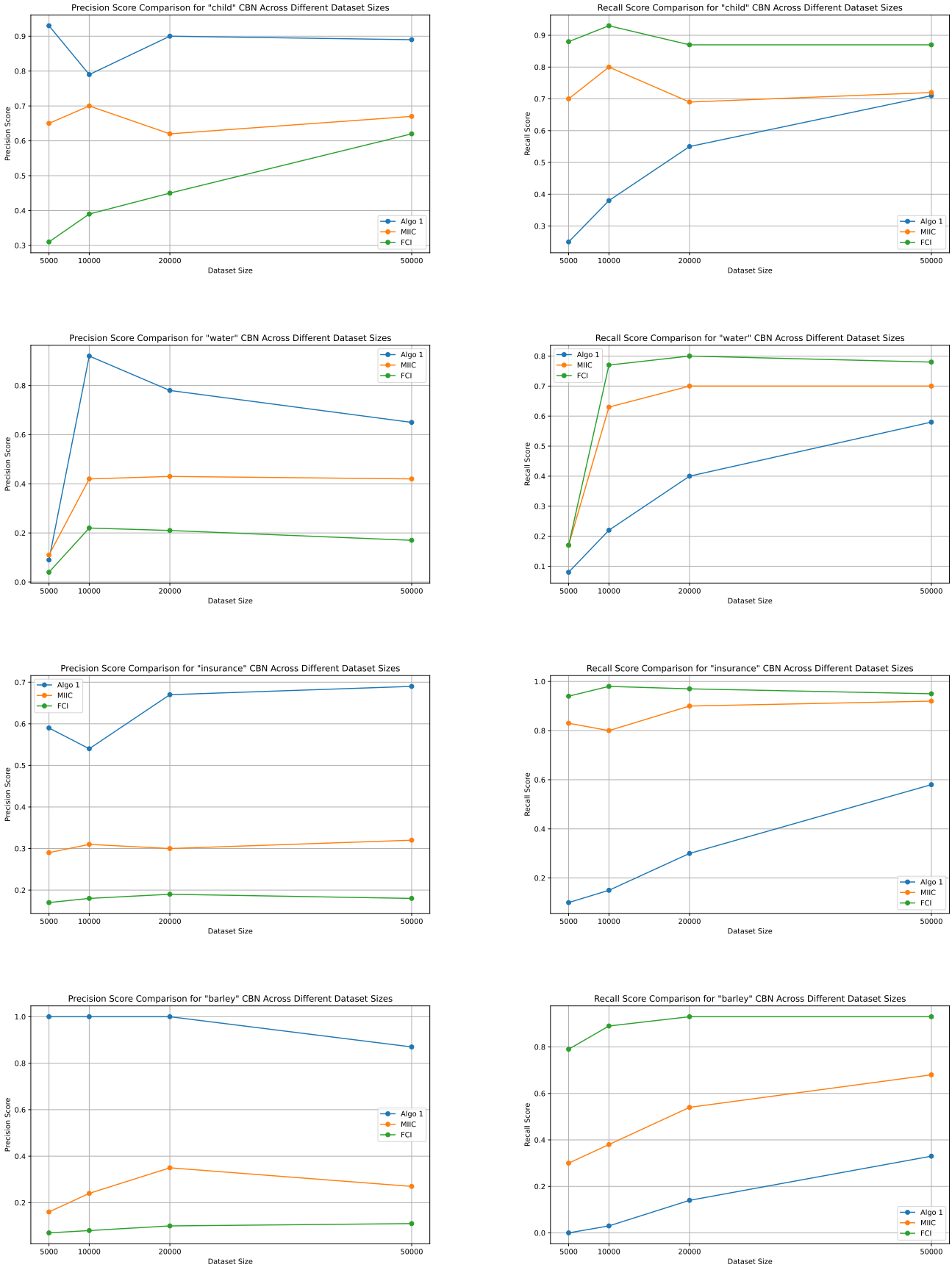


Figure II.12.: Comparative plots for "Precision" and "Recall" metric across different CBNs.

as latent confounders (high ¬ok values), affecting its overall F1 score negatively. These observations are corroborated by the data presented in Figures II.11 and II.12, where FCI's performance is characterized by elevated 'ok' and *recall* metrics. However, it notably lags behind the other methods concerning the '¬ok' and *precision*, as indicated by its substantially higher (for '¬ok') and lower (for *precision*) values in these areas. This discrepancy highlights FCI's relative inefficiency in accurately distinguishing between true confounders and non-confounders, further underscoring the method's propensity for a less discerning identification process.

In contrast, MIIC adopts a more conservative approach, resulting in fewer identified confounders but with greater accuracy, as evidenced by its superior F1 score compared to FCI. Algo 1 refines this conservative approach, achieving the highest *precision* among the three algorithms by making the fewest misidentifications. These results are clearly demonstrated in Figure II.11, where Algo 1 exhibits the lowest '¬ok' values, underscoring its efficacy in accurately identifying true confounders., However its cautious nature also implies a tendency to miss some confounders, resulting in a comparatively lower *recall*, a trend that is evident in Figure II.12. This balance between precision and recall underscores that Algo 1 is yet effective approach to confounder identification.

Notably, the performance of Algo 1 improves with larger datasets, as it correctly identifies an increasing number of confounders, that is also reflected in its improving F1 score. This improvement with larger dataset sizes verifies the empirical implications of Proposition 4.1, suggesting that the algorithm's effectiveness in discerning the true structure of the underlying causal model enhances with more data.

In Table II.3, we compare the CPDAGs derived by Algo 1, MIIC, and FCI against the CPDAGs of the CBNs that generated the datasets, incorporating confounders. To understand the metrics employed in this analysis, refer to Table II.4.

- **ok/miss**: The 'ok' column indicates the arcs and edges accurately identified by each algorithm, mirroring those present in the generating CBNs. Conversely, the 'miss' column enumerates the connections existing in the original CBNs that were not captured in the learned CPDAGs. The table indicates competitive results for Algo 1 and MIIC in terms of fidelity to the original structures. While MIIC occasionally surpasses Algo 1 in identifying correct connections, Algo 1 demonstrates a more pronounced advantage when it leads, particularly in larger datasets, underscoring the effectiveness of our approach in closely approximating the true underlying causal structures.
- **reversed (rev.)**: This column counts the arcs in the CPDAG of the generating CBN that were learned in reverse, suggesting a misinterpretation of the causality direction. FCI tends to have fewer reversed arcs, indicating its strength in this aspect, although the count of such arcs is minimal across all algorithms for various CBNs and dataset sizes.

		Algo 1					**MIIC**					**FCI**				
CBN	$\|\mathcal{D}\|$	ok	miss	rev.	type	xs	ok	miss	rev.	type	xs	ok	miss	rev.	type	xs
child	5000	**22.46**	3.08	1.02	**2.44**	**2.54**	13.42	**2.00**	3.30	10.28	3.42	15.10	4.92	**0.18**	8.80	7.84
	10000	**23.10**	2.52	1.52	**1.86**	**3.12**	13.73	**1.16**	3.73	10.37	3.78	15.64	3.40	**0.06**	9.90	6.00
	20000	**22.88**	1.84	2.22	**2.06**	**3.62**	14.06	**1.38**	3.52	10.04	4.25	16.70	2.60	**0.00**	9.70	4.42
	50000	**22.98**	1.22	2.84	**1.96**	4.34	14.14	**1.20**	3.69	9.96	**4.31**	20.80	1.54	**0.00**	6.66	2.52
water	5000	**16.72**	40.18	4.72	8.38	**4.60**	15.21	**40.15**	7.71	**6.94**	9.50	13.44	45.56	**1.34**	9.66	11.52
	10000	**19.00**	**36.98**	7.58	6.44	**5.96**	17.74	37.78	8.16	**6.32**	9.40	17.92	43.00	**1.06**	8.02	12.70
	20000	21.12	**36.64**	6.82	5.42	**8.74**	**21.88**	38.04	4.92	**5.16**	11.35	17.76	43.78	**0.52**	7.94	16.88
	50000	**26.22**	**32.48**	7.20	**4.10**	**7.04**	23.68	34.70	6.16	5.46	10.64	25.12	38.66	**0.76**	5.46	16.40
insurance	5000	**32.38**	14.44	2.46	**6.72**	**3.74**	29.70	**12.86**	3.51	9.93	12.33	23.94	21.62	**0.44**	10.00	18.10
	10000	**34.68**	12.22	2.38	**6.72**	**4.06**	28.51	**11.57**	3.67	12.24	12.04	25.82	18.56	**0.32**	11.30	17.46
	20000	**38.76**	**9.24**	3.30	**4.70**	**5.04**	29.78	9.68	3.95	12.59	13.51	27.92	16.24	**0.14**	11.70	17.12
	50000	**41.08**	**7.58**	4.42	**2.92**	**6.42**	30.61	8.30	4.04	13.04	13.80	29.46	13.92	**0.10**	12.52	17.84
alarm	5000	38.10	6.42	3.90	1.58	**5.10**	**39.98**	**5.50**	3.28	**1.24**	8.26	37.36	10.34	**0.58**	1.72	10.32
	10000	39.08	4.56	4.92	1.44	**5.84**	**40.96**	**4.48**	3.52	**1.04**	6.98	39.52	8.82	**0.46**	1.20	9.20
	20000	40.64	**3.16**	4.94	1.26	**5.80**	42.41	3.43	3.43	**0.73**	6.27	**42.90**	6.06	**0.18**	0.86	7.60
	50000	40.28	**2.66**	5.82	1.24	7.40	42.45	2.96	3.71	**0.88**	8.20	**43.46**	5.46	**0.18**	0.90	**6.54**
barley	5000	41.46	34.70	2.98	8.86	**6.32**	**49.76**	**25.14**	5.44	7.66	13.88	42.52	38.30	**2.06**	**5.12**	42.42
	10000	46.30	29.14	2.78	9.78	**5.40**	**54.88**	**21.48**	4.76	6.88	13.54	47.64	33.60	**1.32**	**5.44**	40.90
	20000	52.08	22.86	3.58	9.48	**4.64**	**58.53**	**17.78**	5.20	6.49	13.71	53.48	27.38	**1.74**	**5.40**	34.64
	50000	57.74	18.12	4.12	8.02	**4.64**	**64.48**	**12.91**	4.52	6.09	20.93	59.76	21.24	**1.34**	**5.66**	30.92

Table II.3.: Comparisons of the learned CPDAGs with those of the CBNs (with confounders) that generated the datasets.

Metric	**Original Graph**	**Learned Graph**
ok	arc $\longrightarrow$ edge —	arc $\longrightarrow$ edge —
miss	arc $\longrightarrow$ edge —	nothing nothing
reverse	arc $\longrightarrow$ arc $\longleftarrow$	arc $\longleftarrow$ arc $\longrightarrow$
type	arc $\longrightarrow$ edge —	arc — edge $\longrightarrow$
xs	nothing nothing	arc $\longrightarrow$ edge —

Table II.4.: Metrics used in Table II.3

- **type**: This column highlights discrepancies in arc types, specifically where arcs are learned as edges and vice versa, suggesting a mix-up between directed and undirected connections. Algo 1 and MIIC generally perform better than FCI in this regard, with Algo 1 often having a more substantial advantage when it outperforms MIIC.

- **xs**: This column quantifies arcs or edges in the learned CPDAG that do not correspond to any connection between nodes in the generating CBN, indicating spurious causal relationships. Algo 1 excels in this metric, showing a lower propensity to infer incorrect direct causes compared to MIIC and FCI.

For a more intuitive understanding of these comparisons, the corresponding plots, as illustrated in Figures II.13 and II.14, offer a visual representation of these findings, effectively encapsulating the comparative performance of the algorithms in terms of accurately identified connections ('ok'), missed connections ('miss'), changed connections ('type') and spurious connections ('xs'), thereby providing a clearer visualization of their effectiveness in reflecting the original CBN structures."

To assess the robustness of our algorithm in handling more complex scenarios, further experiments were conducted in our published work [Gonzales and Valizadeh 2024], involving increasing number of latent variables and their domain size. As detailed in the supplementary experiments provided in the appendix of our paper, varying the number and domain size of latent confounders showed that the algorithm effectively maintains performance in both structural recovery and latent confounder detection. Moreover, these additional analyses also demonstrate that our method remains stable even under increased complexity constraints, such as allowing a higher number of parents per node.

For full details of these extended robustness analyses, we refer the reader to the appendix of our published paper [Gonzales and Valizadeh 2024].

Overall, Algo 1 demonstrates robust competitiveness, often yielding CPDAGs that more closely resemble the original CBNs than the other evaluated methods, highlighting its efficacy in causal structure learning.

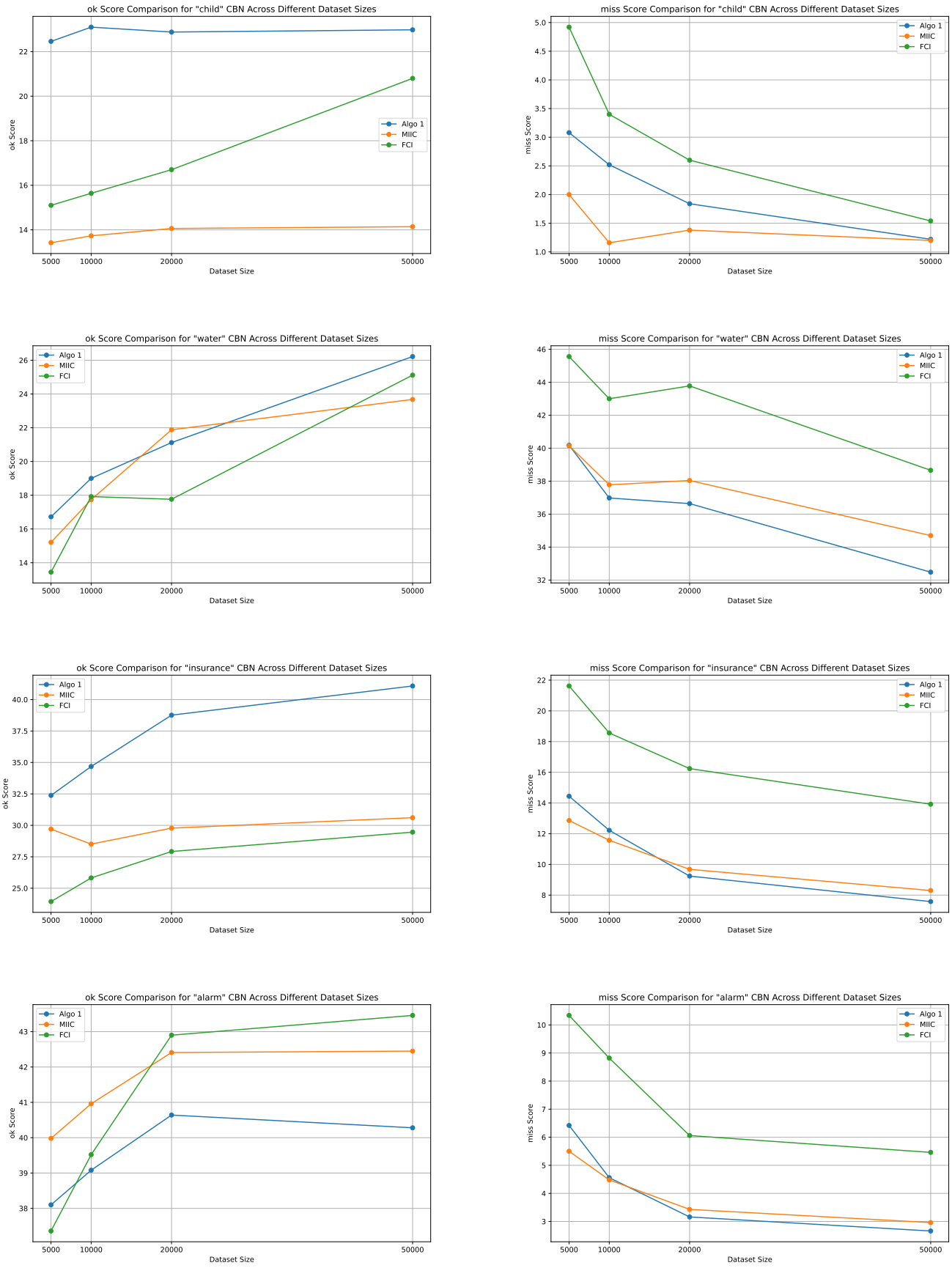


Figure II.13.: Comparative plots for "ok" and "miss" metric across different CBNs' learned CPDAGs.

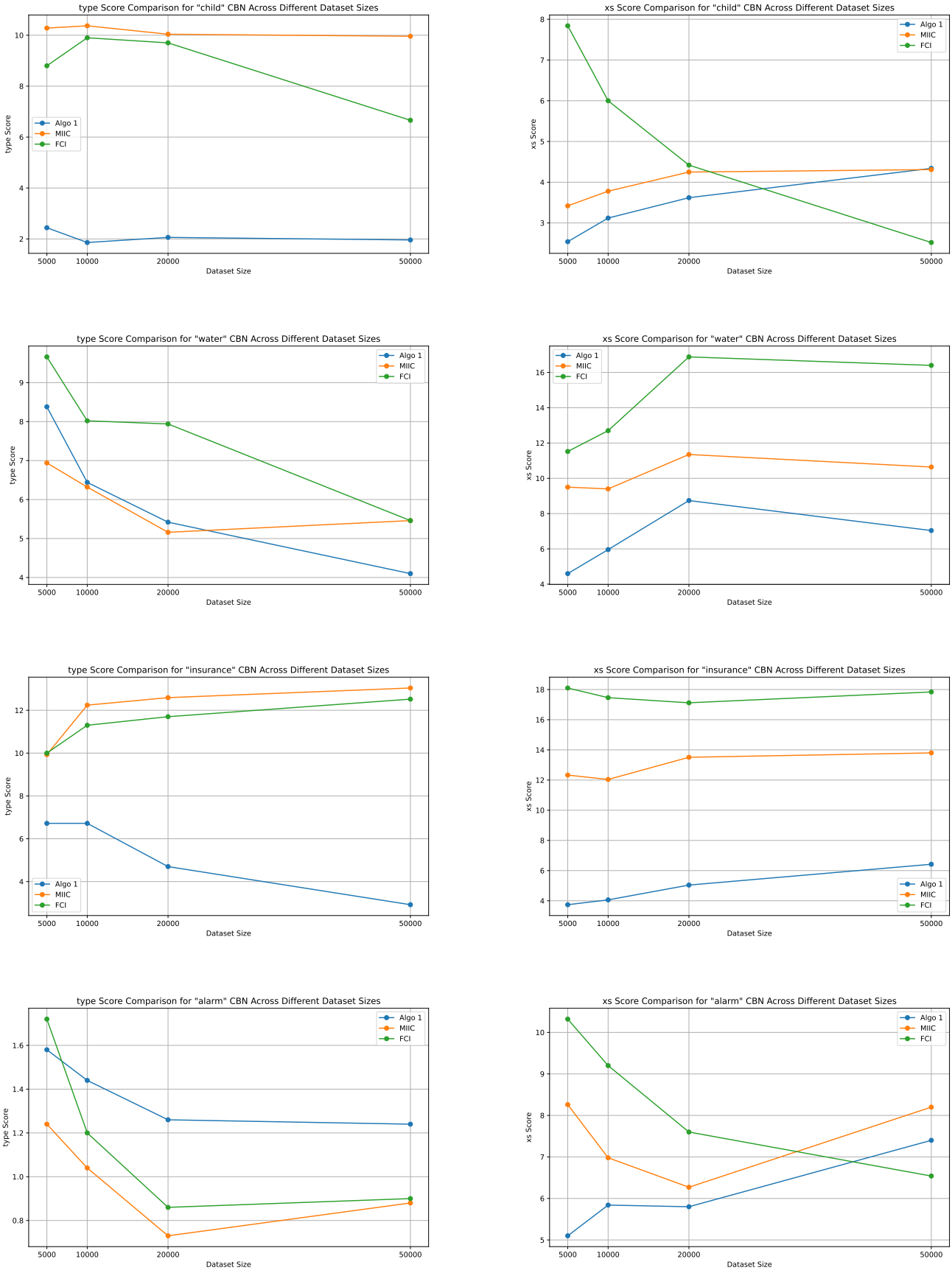


Figure II.14.: Comparative plots for "type" and "xs" metric across different CBNs' learned CPDAGs.

4.4. Conclusion

Throughout this section, we had a comprehensive overview to represent our research's novel methodologies and findings. Beginning with an in-depth analysis of the role of triangles in Directed Acyclic Graphs (DAGs), we delved into the intricate relationship between latent variables and the formation of these geometric structures within the graph. It has been justified mathematically, notably through Proposition 4.1, and theoretically, it is asymptotically guaranteed to produce an I-map. Our exploration of genuine versus latent triangles not only indicated the inherent complexities of causal inference but also prepared us for the innovative algorithmic strategies that followed.

In Section 4.2, we introduced a pioneering score-based Causal Bayesian Network (CBN) learning algorithm for discrete variables designed to adeptly navigate the challenges posed by latent confounders within Bayesian networks. This algorithm, i) exploits only observational data, ii) searches only the space of DAGs, and iii) is capable of identifying latent confounders. Our method is distinguished by its discrimination between genuine and latent triangles, which represents a significant leap forward in our ability to discern the underlying causal structures from observational data. By precisely identifying and modifying the structure based on the detected latent variables, our method enhances the reliability and interpretability of the learned causal models.

The complexity analysis further solidified our understanding of the algorithm's computational efficiency, ensuring our approach remains practical for real-world applications. This theoretical foundation was evaluated through a series of rigorous experiments detailed in Sectio 4.3. Through these empirical studies, we not only validated the efficacy of our algorithm but also demonstrated its superior performance in comparison to existing methodologies.

Our experimental findings underscore the algorithm's remarkable precision (especially for large datasets) in identifying latent confounders, its resilience against spurious connections, and its ability to reconstruct the most accurate and informative causal structures, as represented by CPDAGs. Notably, the results may be judged as more "trustable" than those of its constraint-based competitors like MIIC or FCI in the sense that it makes much fewer mistakes.

In conclusion, this section encapsulates a holistic view of our contributions to causal learning in the presence of latent variables. By addressing critical challenges and proposing robust solutions, we pave the way for more accurate, efficient, and interpretable causal models. Our work not only enhances the theoretical underpinnings of causal inference but also opens new avenues for practical applications in various domains where understanding the causal mechanisms is paramount. As we move forward, we remain committed to refining our methodologies, exploring new frontiers in causal discovery, and contributing to the ever-evolving landscape of data science and artificial intelligence.

5. A Causal Learning Approach for Medical Data

5.1. Introduction

Lung transplant surgeries are critical interventions for patients with end-stage lung diseases, yet they are accompanied by a high risk of mortality and complex post-operative complications. Understanding the intricate web of factors that influence the outcomes of these surgeries is essential for enhancing patient prognosis and management strategies. Bayesian Causal Networks are employed to explore these causal relationships, leveraging their capacity to model complex dependencies.

This section employs the dataset provided by medical experts, detailing variables that encompass patient characteristics, surgical factors, and post-operative outcomes, and discusses specific methodologies employed to derive a causal model that reflects the true dynamics affecting lung transplant surgery outcomes.

The challenge of this exploration is compounded by the limitations inherent in the clinical datasets available, which often include missing values and are limited in size. Such constraints can skew the reliability of the conclusions drawn from statistical analyses. To address the issue of missing data, this study implements the Expectation Maximization (EM) algorithm, associated with Bayesian Networks, to impute missing values with a high degree of precision. It is important to note that the EM algorithm is particularly appropriate in cases where the data is Missing Completely At Random (MCAR), ensuring that the imputation process does not introduce biases associated with the missingness mechanism. By utilizing this approach under the MCAR assumption, we not only restore the integrity of the dataset but also lay the foundation for subsequent analyses based on a more comprehensive representation of the underlying data.

Furthermore, to tackle the challenge posed by the inherently small size of the dataset, we employ a bootstrapping approach, a resampling technique renowned for its efficacy in augmenting small datasets and enhancing the robustness of statistical estimates. Through the generation of multiple simulated samples from the original dataset, bootstrapping enables an exploration of the variability and stability of the inferred network connections, thereby enriching the structural faithfulness of the Bayesian Network model.

Building upon this foundation, this section also integrates the novel score-based approach for latent variable identification developed earlier in the thesis. By incorporating this method into the causal learning process for medical data, we enhance the structural integrity and interpretability of the resulting Bayesian Network. This integration allows us to uncover deeper causal relationships within the medical datasets previously obscured by latent variables.

Central to our methodological framework is the objective to create a reliable and empirically validated network structure that captures the essential causal dynamics influencing surgical outcomes. Through an examination of the consistency and

prevalence of specific structural features across various bootstrap iterations, we aim to construct a Causal Network model that reflects the experimental realities of the dataset and withstands the rigors of statistical validation. By the end of this section, we aim to present a refined causal network model that offers valuable insights into the factors that significantly impact the success of lung transplant surgeries.

5.2. Methodology

5.2.1. Data Description

The dataset provided by medical professionals consists of 311 patient records, each described by a series of variables categorized into three temporal phases relative to the lung transplant surgery. As depicted in Table II.5 these phases include:

1. **Pre-operative variables**: These variables assess general health conditions and include:

- **Age (Age)**: Age of the recipient at the time of transplantation.
- **Sex (Sex)**: Biological sex of the recipient.
- **BMI (Body Mass Index)** and **Height**: Key indicators of physical health.
- **FEV1 (Forced Expiratory Volume in 1 second)** and **FVC (Forced Vital Capacity)**: Measures of lung function.
- **TLC (Total Lung Capacity)**: Total volume of air the lungs can hold.
- **SupEmer, SEm (Super Emergency)**: Indicates the priority of the transplant due to the recipient's critical condition.
- **NIV (Non-Invasive Ventilation Pre-Operatively)** and **O2 (Oxygen Therapy Pre-Operatively)**: Supportive treatments before surgery.

These variables are crucial for assessing patients' overall health and suitability for surgery. (Orders 1 to 4 in Table II.5)

2. **Intra-operative variable**: Details specific to the surgery (Order 5 in Table II.5):

- **Graft_Type, GType (Type of Graft)**: Specifies which lung(s) were transplanted—left, right, or both.

3. **Post-operative variables**: Tracks outcomes and conditions post-surgery, including:

- **MV_Duration, MVD (Duration of Mechanical Ventilation)** and **ECLS (Extracorporeal Life Support Post-Operatively)**: Measures of post-surgical support.
- **AMR (Acute Humoral Rejection Early)**: Indicates early humoral rejection of the graft.

- **BP_1y (Bacterial Infections)**, **VP_1y (Cytomegalovirus pneumonia)** and **MP_-1y (Fungal Infections)**: Occurrences of infections within the first year post-transplant.
- **CLAD (Chronic Lung Allograft Dysfunction)**: A condition affecting long-term graft function and survival.
- **Time_to_Death, TTD (Time to Death or Chronic Dysfunction)**: Tracks survival time post-transplant.

These are indicative of the surgery's success and the patient's recovery trajectory. (Orders 6 to 9 in Table II.5)

Table II.5.: Temporal Order of Variables

Temporal Order	Variables
1	Age, Sex
2	Height
3	BMI, FEV1, FVC, TLC
4	Graft_Listing_Duration (GLD), SupEmer (SEm), NIV, O2
5	Graft_Type (GType)
6	MV_Duration (MVD), ECLS
7	AMR, BP_1y, MP_1y, VP_1y
8	CLAD
9	Time_to_Death (TTD)

5.2.1.1. Data Discretization and Variable Types

Given the small dataset size, data discretization was essential to ensure robust statistical analysis with fewer categories, thus reducing the potential for zero-cell issues in contingency tables. Under the guidance of medical experts, this process aimed to maintain clinical relevance and accuracy. The dataset comprises various types of variables:

- **Binary variables:** Include indicators such as *Sex*, *SupEmer*, *NIV*, *O2*, *ECLS*, *AMR*, *CLAD*, *BP_1y*, and *MP_1y*, which represent yes/no (male/female for *Sex*) options.
- **Categorical variables:** The *Graft_Type* variable indicates the type of lung transplant performed—left (*Gauche*), right (*Droite*), or both lungs (*Bi-pulmonaire*) and is categorized into these three distinct classes.
- **Continuous variables:** These include the rest of the variables containing age, BMI, various medical test results, and time-related measures, which were discretized into meaningful categories to facilitate the analysis. Based on consultations with medical experts, some variables were discretized into two categories and others into three, depending on the clinical significance attributed to each variable.

This categorization and discretization strategy was designed to balance detail with the breadth of analysis possible given the dataset's constraints, ensuring that the causal inferences drawn are both statistically sound and clinically meaningful.

Despite its comprehensive nature, the dataset's limited size and the presence of missing values, which are Missing Completely At Random (MCAR), present challenges. These issues necessitate some statistical techniques to ensure the causal inferences' integrity. The small dataset size, in particular, limits the generalizability of the findings, necessitating careful consideration and validation of the causal relationships identified through the Bayesian Network analysis.

5.2.2. Handling Missing Data

As explained in the previous sections, we have to deal with a dataset with missing values. To impute these missing values, we used Structural Expectation Maximization (EM) in conjunction with Bayesian Networks (BNs), as described by Friedman in his paper on the Bayesian Structural EM algorithm [Friedman 1998].

The Expectation Maximization (EM) algorithm, originating from the foundational work of Dempster, Laird, and Rubin 1977, provides a robust framework for estimating maximum likelihood in cases of incomplete data. Its applicability to our medical dataset is justified, particularly because the missing data are Missing Completely At Random (MCAR). This condition of randomness assures that the missing data points are a random subset of the total data, allowing for the unbiased estimation of parameters. EM sidesteps the direct calculation of the maximum likelihood estimate, which is typically complex or unattainable due to missing data components. Instead, it employs an iterative process that refines parameter estimates and converges toward the maximum likelihood solution by effectively utilizing the available observed data. This iterative improvement is crucial for achieving reliable parameter estimates that mirror the true structure of the underlying data.

The essence of EM lies in its ability to deal with missing data in a manner that is both efficient and theoretically sound. By iteratively alternating between two distinct steps – the Expectation (E) step and the Maximization (M) step – EM effectively infers the missing values and refines the model parameters. During the E step, the algorithm calculates the expected value of the log-likelihood function with respect to the current estimates of the parameters. This step is predicated on the assumption that if the missing data were known, the likelihood of the observed data could be efficiently maximized. Following the E step, the M step seeks to find the parameter values that maximize the expected log-likelihood found in the E step. This process capitalizes on the observed data and the inferred distribution of the missing data to update the parameter estimates, thereby maximizing the likelihood of the observed data under the current model. The alternation between the E and M steps continues iteratively, with each cycle refining the parameter estimates and inching closer to the optimal like-

lihood values. This iterative refinement is the basis of the EM algorithm, embodying a robust approach to unraveling the complexities associated with incomplete datasets.

In our approach, we undertake a crucial step of learning the Bayesian Network (BN) structure before the EM algorithm's iterative process begins. This is achieved by learning a Directed Acyclic Graph (DAG) from a dataset with no missing values, which means we remove all the samples with missing values from the original data. Using the CPBayes exact score-based learning algorithm, as outlined in [Van Beek and Hoffmann 2015], we constructed the DAG for the given dataset. This step involves converting the dataset into entities known as 'instances.' These instances encompass all possible combinations of nodes that CPBayes evaluates for potential parental relationships. To manage the computational demands, limiting the number of potential parents for each node is essential. In our practical tests, we capped this number at four because this limit substantially eases the computational effort, as further discussed in [Van Beek and Hoffmann 2015]. The learned DAG represents the dependencies among the variables in the dataset, essentially capturing the conditional relationships that the EM algorithm will leverage to impute missing data. This learned structure serves as a foundational model for the Expectation (E) step, where the probability distributions and relationships defined by the DAG guide the imputation of missing values.

Expectation (E) Step

In the E step of the EM algorithm, our method generates new data samples based on the current estimates of the BN parameters and the structure learned in the previous step. It calculates the expected value given the observed data and the current BN model for each missing data point. This involves setting up a Bayesian Network with the learned structure and using it to compute the posterior distribution of the missing values given the observed values. The 'LazyPropagation' method from pyAgrum 1.8.3 library [Ducamp, Gonzales, and Wuillemin 2020] is used to perform efficient inference in the BN, enabling the calculation of these posterior distributions.

Maximization (M) Step

Following the expectation step, the M step updates the BN parameters to maximize the likelihood of the observed data given the newly imputed values. We achieve this by learning the parameters of the BN using the complete data (both observed and imputed). The parameters are updated to reflect the distributions that best fit the complete dataset, incorporating the imputed values from the E step. This process iteratively refines the BN model, making it more representative of the underlying data distribution, including the previously missing components.

Iteration and Convergence

The method iterates between the E and M steps, each time refining the imputed values and the model parameters. This iterative process continues until the BN structure no longer changes, indicating convergence. It employs a convergence criterion

based on the stability of the BN structure across iterations, ensuring that the algorithm terminates once the imputed values have stabilized.

Algorithm 4 shows the detailed steps of the EM approach for imputing the missing values using a first-order approximation [Friedman 1998].

Algorithm 4: EM Algorithm for Missing Data Imputation using Bayesian Networks

Input: Data with missing values ($\mathscr{D}_0$)
Output: Imputed Data and corresponding structure

$\mathscr{D} \leftarrow$ Drop samples with missing values from $\mathscr{D}_0$;
$\mathscr{G} \leftarrow$ Learned DAG from $\mathscr{D}$ using CPBayes;
while *not converged* **do**
    /* Expectation (E) Step */
    $\mathscr{B} \leftarrow$ Bayesian Network (BN) with current parameters (from $\mathscr{D}$) and structure ($\mathscr{G}$);
    $\mathscr{D}' \leftarrow \mathscr{D}_0$;
    for *each i = instance with missing value(s) in* $\mathscr{D}'$ **do**
        Identify the observed variables and missing data points;
        Compute the posterior distribution for the missing value(s) given the observed data using 'LazyPropagation' on $\mathscr{B}$;
        inst $\leftarrow$ Imputed new instances based on the computed posterior distribution;
        $\mathscr{D}' \leftarrow \mathscr{D}' +$ inst;
        Remove i from $\mathscr{D}'$
    /* Maximization Step */
    $\mathscr{G}_t \leftarrow$ new DAG learned by CPBayes on $\mathscr{D}'$;
    /* Check for Convergence */
    if $\mathscr{G}_t = \mathscr{G}$ **then**
        convergence $\leftarrow$ True
    else
        $\mathscr{G} \leftarrow \mathscr{G}_t$
    if *convergence* = *True* **then**
        Break
return $\mathscr{D}', \mathscr{G}_t$

5.2.3. Bootstrapping for Data Augmentation

Having addressed the issue of missing values within the dataset, we encountered another significant challenge: the limited size of the dataset, which poses constraints on the reliability of statistical inference and the robustness of learned models. To mitigate the limitations imposed by the dataset's small size, we employed a bootstrapping

approach, a well-established resampling technique that enhances the estimation of population statistics from small or incomplete datasets. This methodology, particularly the framework and its application to Bayesian Network structure learning, draws upon foundational ideas discussed in Friedman et al. Friedman, Goldszmidt, and Wyner 2013.

Bootstrapping involves generating multiple simulated samples (bootstrap samples) from the original dataset by sampling with replacement. This technique is particularly advantageous in scenarios characterized by limited data, as it facilitates the assessment of the variability and reliability of statistical estimates, thereby providing a more robust foundation for inferential statistics. The essence of bootstrapping lies in its ability to approximate the distribution of a statistic across numerous resampled datasets, thereby enabling a deeper understanding of the statistic's stability and variability in the context of the broader population.

In the context of our study, the primary objective was to find the true structure of the Causal Bayesian Network representing the relationships among the variables in our dataset. To achieve this, we focused on accurately identifying the correct connections (e.g., arcs and edges) within the learned Completed Partially Directed Acyclic Graphs (CPDAGs). Consider a network structure, G. A feature of interest f within this structure might be the presence of an arc $X \to Y$ or an edge X—Y, or presence of any connection between two nodes within the CPDAG associated with G. These features can generally be represented as functions mapping network structures to the set $\{0,1\}$. The bootstrapping method employed in our analysis can be articulated through the steps depicted in Algorithm 5:

Algorithm 5: Bootstrapping for Bayesian Network Structure Learning

Data: Original dataset D with N samples
Result: Estimated feature $\bar{f}$ representing the structure of interest

for $i \leftarrow 1$ **to** n **do**
    $D_i \leftarrow$ Resampled N observations with replacement from D
    $G_i \leftarrow$ Learned DAG structure from D_i
    $M_i \leftarrow$ CPDAG of G_i
    $f(M_i) \leftarrow$ Desired structural feature (e.g., arcs, edges) of M_i
$\bar{f} = \frac{1}{n}\sum_{i=1}^{n} f(M_i)$
return $\bar{f}$

This bootstrapping approach not only provided a means to assess the variability and reliability of the learned Bayesian Network structures but also offered insights into the stability of specific structural features across the resampled datasets. By aggregating the structural features across all CPDAGs, we were able to derive a robust estimate of the true network structure, thereby overcoming the limitations posed by the small size of the original dataset.

Our bootstrapping methodology bears similarity to the approach proposed by Nguyen, Leray, and Ramstein 2011, who also applied bootstrap resampling to Bayesian Network structure learning. However, an important distinction lies in the aggregation phase: Nguyen et al. employ a different strategy for aggregating learned structures, particularly in their treatment of V-structures. In contrast, our approach explicitly focuses on aggregating structural features (such as arcs and edges) by analyzing their frequency of occurrence across bootstrap samples, which provides a direct empirical measure of confidence. This strategy facilitates a clear interpretation and validation of the robustness of each structural feature.

Furthermore, the bootstrapping methodology employed in our study extends beyond the mere augmentation of the dataset; it inherently presents a quantifiable measure of confidence in the identified features of interest within the learned CPDAGs corresponding to the data. By examining the frequency and consistency of specific structural features across the multitude of CPDAGs generated in each bootstrap iteration, we are able to assess the robustness and reliability of these features. This frequency of occurrence serves as an empirical factor for the confidence level associated with each feature, including the strength of evidence for the presence of particular arcs and edges in the true network structure.

Such a statistical approach enables us to identify and validate the significance and stability of structural features within the Bayesian Network, thereby enhancing the credibility and interpretability of the inferred network model. The iterative resampling and analysis supports our findings with a level of confidence that is essential for drawing substantive and reliable conclusions from the study, especially in the face of dataset limitations and inherent uncertainties.

5.2.4. Constructing a Trustworthy Structure

To seamlessly integrate the methodologies outlined in the preceding sections, our subsequent phase synthesizes the Expectation Maximization (EM) algorithm for handling missing data with the bootstrapping technique for data augmentation. This fusion is aimed at consolidating the strengths of both approaches to establish a comprehensive dataset conducive to learning a robust Bayesian Network structure.

Between bootstrapping iterations, the Expectation Maximization (EM) algorithm was applied to each bootstrap sample to address missing data within the dataset. This combined application ensured that each bootstrap sample was not only representative of the original dataset's diversity but also complete, with missing values imputed to maintain the integrity of the data for subsequent analysis. The EM algorithm, renowned for its efficacy in handling missing data within statistical models, iteratively estimates the missing values based on the current model's parameters and the observed data. This iterative process continues until the parameter estimates converge to a stable solution, ensuring that the imputed values are consistent with the underlying data distribution.

Algorithm 6: Parallel EM Imputation and Bootstrapping for CPDAG Generation

Input: Original dataset D_0 with missing values, Number of bootstrap samples n

Output: Set of imputed bootstrap samples $\{D_i^*\}_{i=1}^n$, Set of CPDAGs $\{M_i\}_{i=1}^n$

for $i \leftarrow 1$ **to** n **do**
- $D_i \leftarrow$ Create a bootstrap sample by resampling D_0 with replacement
- `/* Apply EM algorithm to the bootstrap sample and learn the DAG structure from it */`
- $G_i, D_i^* \leftarrow$ Outputs of Algorithm 4 on D_i
- `/* Convert DAG to CPDAG */`
- $M_i \leftarrow$ Convert G_i into a CPDAG
- Store M_i for further analysis

return *Set of imputed bootstrap samples* $\{D_i^*\}_{i=1}^n$, *Set of CPDAGs* $\{M_i\}_{i=1}^n$

Algorithm 6 outlines the procedure for performing EM-based imputation in parallel with generating bootstrap samples. Each sample undergoes the EM imputation process, ensuring that the resultant datasets are complete and ready for learning BN structures. This comprehensive approach enhances the reliability of the inferred network, as it is constructed from a dataset enriched through bootstrapping and refined through precise imputation of missing values.

Upon successfully implementing the bootstrapping and EM approach for generating CPDAGs, our subsequent endeavor was to utilize the accumulated results to derive a dependable structure for our network. Central to this process was the calculation of confidence levels for each potential connection within the network, encompassing both arcs and edges. This confidence level, determined through the frequency of occurrence across various bootstrap iterations, served as an essential metric in discerning the structural integrity and reliability of the connections. In addition to the arcs and edges in a CPDAG, we introduce a broader group of connections called **links**. A link in a CPDAG refers to any connection (an arc or an edge) between two nodes. In other words, if two nodes are connected in the skeleton of the graph, we say there is a link between them.

To systematically integrate these links into our final structure, we established a predefined threshold of confidence. Links surpassing this threshold were deemed robust and consistent enough to be incorporated into the ultimate structure. This selection process ensured that only the most statistically significant connections, as evidenced by their prevalence across multiple bootstrap samples, were included in the final model. Consequently, this approach not only enhanced the structural fidelity of our network but also endowed it with a higher degree of empirical validation. Algorithm 7 illustrates the procedure of producing such a structure.

Algorithm 7: Algorithm for Deriving Trustworthy Structure from Bootstrap Results

Data: Primary learned CPDAG M_0 using the original data, Set of CPDAGs $\{M_i\}_{i=1}^{n}$ obtained from bootstrap iterations, a *threshold* for links' occurrences to consider them as being confident

Result: A CPDAG with confident connections

$\{S_i\}_{i=0}^{n} \leftarrow$ skeletons of $\{M_i\}_{i=0}^{n}$
$\mathcal{V}_0 \leftarrow$ set of all v-structures in M_0
$\mathcal{V}_C \leftarrow \emptyset$ `// Set of confident v-structures in` M_0
$\mathcal{L} \leftarrow$ Set of all links in $\{S_i\}_{i=1}^{n}$
$\mathcal{L}_0 \leftarrow$ Set of all links in S_0
$\mathcal{L}_C \leftarrow \emptyset$ `// Set of confident links`
for *each* $l \in \mathcal{L}$ **do**
    $f_l \leftarrow$ number of l's occurrences in $\{S_i\}_{i=1}^{n}$
    $c_l = \frac{f_l}{n} \times 100$ `// Confidence level`
    if $c_l >$ *threshold* **then**
        $\mathcal{L}_C \leftarrow \{l\} \cup \mathcal{L}_C$
for *each v-structure* $V \in \mathcal{V}_0$ **do**
    $l_1, l_2 \leftarrow$ the links of V
    if $l_1 \in \mathcal{L}_C$ *and* $l_2 \in \mathcal{L}_C$ **then**
        $\mathcal{V}_C \leftarrow \{V\} \cup \mathcal{V}_C$
for *each* $l \in \mathcal{L}_0$ **do**
    if $l \notin \mathcal{L}_C$ **then**
        Remove l from S_0
Orient the links of S_0 which are members of $\mathcal{V}_C$
Create a CPDAG of S_0 using Meek's rules

This methodology of creating a reliable structure from the bootstrap results shows the robustness of the connections included in the final structure and introduces a quantifiable measure of confidence in the network's representation of the underlying data relationships. Such an empirically grounded approach can be very helpful in the realm of data-driven modeling, particularly in scenarios where the available dataset is constrained in size or completeness.

5.2.5. Weighted Causal Effects

Understanding the relationships between various health determinants and outcomes is very important in the complex field of medical data analysis. Medical specialists and researchers are often faced with the challenge of deciphering complex causality networks, where multiple factors may converge to influence a single health outcome.

This complexity is compounded by the inherent variability and interconnectedness of medical data, where the impact of individual characteristics can be subtle yet significant in the aggregate.

While the structure of a Bayesian Network can illustrate potential causal pathways, the critical task remains to quantitatively assess the influence or importance of each contributing factor, especially when multiple causes converge on a common effect. This need for a quantitative measure becomes even more pronounced when considering the practical implications of such analyses.

Medical experts rely on these insights to make informed decisions about patient care strategies and public health interventions. Hence, there's a pressing need for a methodological approach that highlights the presence of causal relationships and quantifies the intensity of each cause's impact on the outcome.

5.2.5.1. Shapley Values: Bridging the Quantitative Gap

To address this gap, we turned to the concept of Shapley values, a well-established method from cooperative game theory known for its equitable distribution of payoffs based on individual contributions. Originating from Lloyd Shapley's work in 1953 [Shapley 1953], the appeal of Shapley values lies in their ability to offer a fair and comprehensive quantification of each factor's marginal contribution, considering all possible combinations of contributing causes.

In the context of our research, utilizing Shapley values to weigh the arcs in the learned Bayesian Network structure emerged as a promising approach [M. H. Ali, Le Biannic, and Wuillemin 2021]. By applying these values to the scores derived from the data using a score-based learning algorithm (CPBayes), we aimed to provide a quantitative measure of the influence exerted by each parent node (cause) on their common child node (effect) within the network.

This motivation was driven by the dual need to satisfy the domain experts' demand for actionable insights and to enhance the interpretability of the BN model. By quantifying the intensity of different causes on a common effect, we aspire to transcend the qualitative boundaries of network structures, offering a more profound, data-driven understanding of causal dynamics in medical data. This integration was aimed at weighing the arcs, thereby providing a nuanced understanding of the relative importance of parent nodes to their respective child nodes within the network. This quantitative assessment, rooted in the equitable principles of Shapley values, promises to transform complex network relationships into tangible insights, empowering medical professionals and researchers with precise, evidence-based knowledge to inform their decisions.

The Shapley value for player i in a game with players N is defined as:

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} (v(S \cup \{i\}) - v(S)) \tag{II.5}$$

where:

- N is the set of all players.
- S is a subset of N not containing player i.
- $|S|$ is the number of players in subset S.
- $|N|$ is the total number of players.
- $v(S)$ is the value (or payoff) that the coalition S achieves.
- $v(S \cup \{i\}) - v(S)$ represents the marginal contribution of player i when joining the coalition S.

The formula calculates the average marginal contribution of player i across all possible coalitions S that they can join.

In the context of Bayesian Networks, the "players" in the cooperative game are analogous to the parent nodes of a common child node in the network. The "value" or "payoff" ($v(S)$) is represented by the scores derived from the CPBayes algorithm, which quantify the strength or influence of a set of parent nodes on the child node.

To adapt the Shapley value calculation to this context, we utilize Equation II.5, where we redefine N, S, and $v(S)$ as follows:

- N as the set of all parent nodes for a given child node in the Bayesian Network.
- S as a subset of parent nodes being considered.
- $v(S)$ as the score associated with the child node given the parent nodes in set S, as determined by the CPBayes algorithm.

The Shapley value for a parent node i (ϕ_i) quantifies the average impact of adding node i to different combinations of parent nodes on the score of the child node.

Here, $v(S \cup \{i\}) - v(S)$ is the difference in the CPBayes score for the child node when parent node i is added to the set S of other parent nodes. This difference represents the marginal contribution of the parent node i to the child node's score, accounting for all possible combinations of other parent nodes.

By computing the Shapley value for each parent node in relation to a specific child node, we obtain a fair and comprehensive measure of each parent's influence on the child, based on the CPBayes-derived scores. This approach allows us to quantitatively assess the importance of each arc (parent-child relationship) in the learned BN structure, providing valuable insights into the causal dynamics captured by the network.

5.2.5.2. Graphical Representation of Shapley Values

Upon finalizing the structure of the Bayesian Network as represented by the CPDAG, a critical step in our analysis involved weighing all directed edges (arcs) within the network. This process was underpinned by the calculated Shapley values, which served as a quantitative metric to assess the relative influence of parent nodes on their respective child nodes. Each directed arc in the CPDAG was assigned a weight that reflected the magnitude of this influence, based on the Shapley value computations as outlined in Equation II.5.

It is important to note that the CPDAG may also contain undirected edges, indicative of relationships where the learning algorithm does not definitively establish the directionality (and hence causality) between nodes. In the context of our analysis, these undirected edges were not subjected to the weighing process. The absence of clear causal directionality precludes the meaningful application of Shapley value-based weighing, as the foundational premise of Shapley values—in assessing the contribution of a cause to an effect—relies on a directed causal framework. Consequently, undirected edges in the CPDAG were retained without assigned weights, reflecting the ambiguity in their causal interpretation within the network structure.

This approach ensures that our analysis remains firmly grounded in causal principles, providing a clear and quantitatively grounded insight into the causal dynamics encapsulated by the Bayesian Network. By doing so, we adhere to a rigorous analytical framework that distinguishes between definitive causal relationships and those that remain undetermined, thereby enhancing the interpretability and applicability of our findings in medical data analysis.

An integral aspect of our analysis involved the intuitive visualization of the Bayesian Network, specifically through the graphical representation of Shapley values as the thickness of the arcs connecting parent nodes to child nodes. This visualization technique was implemented to enhance the interpretability of the network's causal dynamics, providing a clear and immediate visual cue regarding the strength of influence exerted by each parent node.

A customized Python script facilitated the process of mapping Shapley values to arc thickness. For each arc in the network, the corresponding Shapley value determined the width attribute in the DOT language used for graph visualization, with higher Shapley values resulting in greater thicknesses. This mapping was applied exclusively to directed edges within the network, reflecting our focus on quantifiable causal influences.

In addition to arc thickness, the script also allowed for the differentiation of arcs by color, further enhancing the graphical representation's ability to convey information. Directed arcs were colored based on the child node they pointed to, providing a visual grouping of parent nodes influencing a common child. In contrast, undirected edges, representing relationships without a determined causal direction, were uniformly

depicted with a default thickness and color (black), underscoring their distinct status within the CPDAG.

This approach to visualization not only aids in the qualitative understanding of the network's structure but also embeds a layer of quantitative information directly into the graphical representation. By doing so, the graphical structure transcends its role as a mere schematic of relationships and becomes an informative portrait that communicates the interplay of influences within the network, grounded in the rigorous computation of Shapley values.

To concretize our discussion on applying Shapley values within Bayesian Networks, let us consider a specific example illustrated in Figure II.15. The figure showcases a part of a learned structure based on medical data where each arc connecting parent nodes to a child node is not only color-coded to denote different child nodes but also varies in thickness to reflect the Shapley values calculated for each parent node's influence. Table II.6 represents the corresponding Shapley values for these arcs.

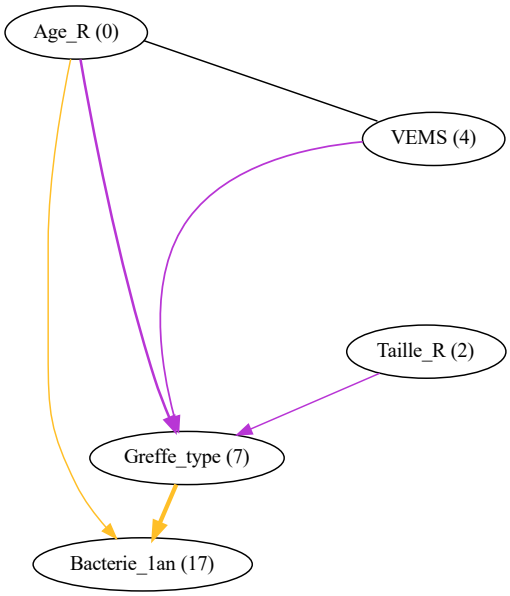


Figure II.15.: Network with Weighted Arcs

Child	Parents			
-	0	2	4	7
7	18.3	10.72	12.67	-
17	2.56	-	-	7.45

Table II.6.: Raw Shapley Values of the Network in Figure II.15

While the computed Shapley values provide a robust quantitative measure of each parent node's relative influence, their direct application as graphical arc thicknesses presents a challenge. The raw Shapley values often span a wide range, resulting in a graphical representation that can be visually overwhelming and less informative. Extremely high values could lead to disproportionately thick arcs, while smaller values might render some arcs nearly invisible, thereby diminishing the graphical structure's interpretability.

To address the challenges in graphical representation, a more nuanced scaling approach was implemented to normalize the Shapley values, ensuring a balanced and visually appealing visualization. This method was particularly attentive to the varying range and distribution of Shapley values across different parent node sets influencing a common child node. For sets with a single parent node, the arc's width was set to a predefined minimum width, ensuring visibility and consistency in the graphical representation. For sets with multiple parent nodes, the scaling was dynamically adjusted based on the ratio of the maximum to minimum absolute Shapley values among the parents. A threshold ratio (`ratio_limit`) was established to determine the scaling strategy: if the max/min ratio was within this limit, the scaling was based on the minimum Shapley value, ensuring that the smallest value was represented by the minimum width (`min_width`), and other values were scaled proportionally. However, if the max/min ratio exceeded this threshold, indicating a significant disparity in Shapley values, the scaling was adjusted so that the maximum value corresponded to a maximum width (calculated as `ratio_limit` × `min_width`), with other values scaled down accordingly. This approach allowed for a more equitable representation of Shapley values, ensuring that each arc's thickness was both visually discernible and proportionate to its relative influence, as quantified by the Shapley value. Table II.7 represents the scaled Shapley values of Table II.6.

Child	Parents			
-	0	2	4	7
7	1.71	1	1.18	-
17	1	-	-	2.91

Table II.7.: Normalized Shapley Values of the BN in Figure II.15

In the context of our normalization approach, it is imperative to clarify the scope and limitations of comparability among these normalized values. Due to the intrinsic nature of the normalization process, which is conducted individually for each set of parent nodes associated with a specific child node, the resulting normalized values—or correspondingly, the thicknesses of arcs in the graph—are tailored to elucidate the relative influence of parent nodes within that specific parent-child context. It is crucial to acknowledge that equal normalized values or arc thicknesses for incoming arcs to different child nodes do not imply equivalence in their raw Shapley values nor denote a comparable level of causal effect exerted by the parent nodes. The

normalization procedure scales the Shapley values based on the distribution within each parent node set for a child, thereby rendering the normalized values contextually bound to their respective parent-child subsets. Therefore, when interpreting the Bayesian Network's graphical representation, arc thickness comparisons should be confined to arcs sharing a common child node. This approach ensures a meaningful assessment of the relative importance of parent nodes in influencing a particular child node. Attempting to compare the thickness of arcs across different child nodes would overlook the distinct scaling factors applied during normalization, leading to potential misinterpretations of the causal dynamics depicted by the network.

This scaling approach ensures that all contributing parent nodes remain visually discernible in the network structure, with arc thicknesses proportionally representing their relative influence based on Shapley values. By calibrating the graphical representation in this manner, we preserve the quantitative insights offered by Shapley values while enhancing the graphical structure's aesthetic coherence and interpretability.

5.2.6. Latent Variables Identification

After constructing the initial causal network structure, we proceed to examine any triangle configurations within the graph. This examination utilizes the approach introduced in Section 4.1.1, which is designed to discriminate between genuine and latent triangles. In instances where latent triangles are identified, we modify the graph to correct potentially spurious connections, thereby refining the network to represent the underlying causal mechanisms better. This step is crucial for ensuring that the network does not merely depict observable associations but aligns closely with the true causal relationships implied by the data.

5.3. Experiments

The core objective of our experimental investigation was to ascertain the efficacy of a bootstrap-based method in learning Bayesian Network (BN) structures from limited datasets. Given the intrinsic challenges posed by small sample sizes—primarily overfitting and poor generalizability—we specifically designed our experiments to address these issues. Our dataset comprises only 311 samples, which significantly constrains the learning process and necessitates innovative approaches like bootstrapping to enhance data robustness.

5.3.1. Model Generation

Classical BN models, such as Child ($|\mathbf{X}| = 20$), Water ($|\mathbf{X}| = 32$), Insurance ($|\mathbf{X}| = 27$), Alarm ($|\mathbf{X}| = 37$), and Barley ($|\mathbf{X}| = 48$), sourced from the renowned BNLearn Bayesian network repository[3], served as benchmarks for our study. These models provide a

[3] https://www.bnlearn.com/bnrepository

diverse array of structures and parameters, making them ideal candidates for testing the effectiveness of our methods against the limitations posed by the small dataset size.

Our experimental design was structured in two steps: bootstrap sampling for dataset augmentation and Bayesian structure learning for inferring BN structures. Initially, we generated multiple bootstrap samples from each classical BN model's dataset, employing resampling with replacement. This technique not only enriched our dataset but also introduced the necessary variability for robust analysis. By leveraging the power of pyAgrum 1.8.3 library [Ducamp, Gonzales, and Wuillemin 2020], we generated high-quality data for our experiments with ease. This robust tool is specifically designed for modeling Bayesian networks, making it the ideal choice for our research.

Subsequently, we engaged in Bayesian structure learning for each bootstrap sample, facilitated by an exact score-based algorithm for structure learning. This approach adapted the bootstrap samples for CPBayes algorithm to deduce the BN structure from these samples as detailed in [Van Beek and Hoffmann 2015]. This process was iteratively performed across 1000 bootstrap samples for each BN model, ensuring a comprehensive evaluation.

5.3.2. Validation of Bootstrapping Technique

In the pursuit of optimizing the structure learning process as described in previous sections, particularly in the context of bootstrapping, a critical component of our methodology involved the specification of a threshold value (Algorithm 7). This threshold determines the inclusion of connections in the final learned structure based on their frequency of appearance across multiple bootstrapped datasets as explained in Section 5.2.4.

To identify the most suitable threshold value, we embarked on a series of empirical tests encompassing a wide range of threshold percentages. The aim was to discern a value that effectively balances the trade-off between capturing genuine connections present in the original Causal Bayesian Network (CBN) models (ground truth) and minimizing the inclusion of spurious connections.

Through rigorous experimentation across various models and scenarios, we observed a consistent pattern: a threshold set at 60 percent emerged as the optimal choice. This empirical finding indicates that connections appearing in more than 60 percent of the iterations tend to be those that are indeed present in the generating CBN models, reflecting the true underlying causal relationships. Conversely, connections appearing less frequently are more likely to be artifacts of the bootstrapping process or noise in the data rather than genuine causal links.

The decision to set the threshold at 60 percent is thus grounded in empirical evidence gathered from our extensive testing. This threshold effectively filters out

CBN	Original		Score	Output	ok		miss		rev.	type		xs		links		
	Arcs	Edges			Arcs	Edges	Arcs	Edges	Arcs	E2A	A2E	Arcs	Edges	T	F	M
Child	13	12	AIC	Primary	8	3	2	2	1	7	2	18	0	21	18	4
				Bootstrap	5	7	3	3	0	2	5	0	0	19	0	6
			BIC	Primary	3	9	4	2	0	1	6	1	0	19	1	6
				Bootstrap	3	8	4	4	0	0	6	0	0	17	0	8
Insurance	34	18	AIC	Primary	18	2	13	6	2	10	0	11	0	32	11	19
				Bootstrap	17	1	15	8	0	9	2	2	0	29	2	23
			BIC	Primary	5	4	19	10	0	4	10	2	3	23	5	29
				Bootstrap	4	7	20	11	0	0	10	0	0	21	0	31
Water	60	6	AIC	Primary	9	0	43	0	5	6	3	23	2	23	25	43
				Bootstrap	2	5	47	0	0	1	11	2	0	19	2	47
			BIC	Primary	2	6	47	0	0	0	11	2	2	19	4	47
				Bootstrap	0	6	48	0	0	0	12	0	0	18	0	48
Alarm	40	8	AIC	Primary	30	0	5	1	5	4	2	25	1	41	26	6
				Bootstrap	25	3	6	1	2	1	9	0	1	40	1	7
			BIC	Primary	16	1	10	2	5	1	11	0	2	34	2	12
				Bootstrap	15	1	15	2	5	1	7	0	1	29	1	17
Barley	71	13	AIC	Primary	13	1	41	5	5	3	16	10	5	38	15	46
				Bootstrap	9	1	46	6	1	2	19	4	2	32	6	52
			BIC	Primary	0	2	53	11	0	0	18	0	7	20	7	64
				Bootstrap	0	2	54	11	0	0	17	0	6	19	6	65

Table II.8.: Comparisons of the primary and the bootstrapping learned structures, with the Original CBNs that generated the datasets.

less reliable connections while retaining those with substantial support across bootstrapped iterations, thereby enhancing the fidelity of the learned structures to the ground truth models. It ensures that the final causal graphs derived from our experiments are both accurate and trustworthy, which is particularly crucial in applications where precise causal inference is paramount, such as in medical research and policy-making.

The execution of this methodology across different classical BN models provided us with a rich dataset of experimental results. These results were meticulously recorded, capturing key metrics like what we had in Table II.4 such as the number of correct edges ('ok'), missing edges ('miss'), reversed edges ('rev.'), extra edges ('xs'), and we introduce some new metrics: wrongly learned types('type') including edges wrongly learned as arcs (E2A) and arcs wrongly learned as edges (A2E), the total number of learned connections ('Links') that are correctly (T), wrongly (F), and missed (M) in the structure which are calculated based on the values of other columns in the dataset. Specifically, the calculations are as follows:

- links (T) = ok (Arcs) + ok (Edges) + rev. + type (E2A) + type (A2E)

- links (F) = xs (Arcs) + xs (Edges)

- links (M) = miss (Arcs) + miss (Edges)

The metrics were evaluated under two distinct conditions: 'Primary,' using the original dataset generated from the BN model without bootstrap sampling, and 'Boot-

strap,' employing our bootstrap-based method. Table II.8 shows the results of this experiment in detail.

A preliminary overview of the results illustrates the impact of bootstrapping on structure learning in Bayesian Networks. While bootstrapping does not uniformly improve all aspects of the learned structures, its significant reduction in false connections emerges as a crucial benefit. In the domain of medical data analysis, where the accuracy and trustworthiness of inferred causal relationships are paramount, the ability of bootstrapping to filter out spurious connections without substantially increasing the omission of true connections is particularly valuable.

To facilitate a clear and intuitive interpretation of these results, we employed graphical analysis, creating a series of bar plots for each metric across the different BN models. These plots effectively illustrated the comparative performance of the 'Primary' and 'Bootstrap' methods, highlighting the improvements and changes brought about by the bootstrap approach.

By investigating the results from Table II.8 and provided Figures in the context of Causal Bayesian Network (CBN) structure learning using bootstrapping, we observe several key insights. The primary objective of employing bootstrapping in this experiment was to ascertain whether a model learned through multiple bootstrapped datasets could yield a more reliable and accurate representation of the underlying causal structure, particularly in scenarios involving small datasets, which is common in medical data analysis.

As we can see in Figure II.16, the 'ok (Arcs)' metric did not show significant improvement with bootstrapping, suggesting that the identification of direct causal relationships between variables remains challenging even with multiple resampled datasets. Conversely, 'ok (Edges)' demonstrated accepting enhancements in some models with bootstrapping, indicating a marginal improvement in recognizing undirected relationships.

Both 'miss (Arcs)' and 'miss (Edges)' metrics exhibited minimal changes with bootstrapping as it is shown in Table II.8 and represented by links (M) metric as well (Figure II.19b). This suggests that bootstrapping, while helpful, may not substantially reduce the omission of true connections in the learned structures. In the table we observe that the 'rev.' metric, representing the number of arcs learned in a reversed orientation, improved with bootstrapping in some models. This improvement indicates that bootstrapping can help correct some directional errors in the learned causal relationships.

The 'type (E2A)' and 'type (A2E)' metrics as depicted in Figure II.17, indicating errors in classifying edges as arcs and vice versa, saw improvements with bootstrapping. This suggests that bootstrapping contributes to a more accurate distinction between direct and indirect or confounded relationships.

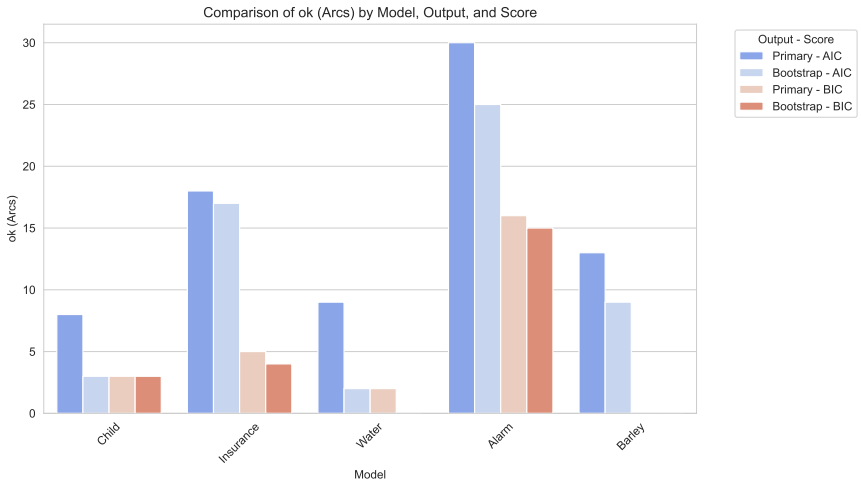
Figure II.18 illustrates a significant finding from the experiment, that is the notable reduction in 'xs (Arcs)' and 'xs (Edges)' metrics with bootstrapping. This reduction in falsely learned connections, particularly evident in the 'links (F)' metric (Figure II.19c), highlights the primary advantage of bootstrapping. By minimizing the inclusion of spurious connections in the learned structure, bootstrapping enhances the reliability and trustworthiness of the model.

The slight improvements in recognizing true connections (as seen in 'ok (Edges)') and correcting some directional errors ('rev.') further complement this primary benefit. These enhancements, coupled with the substantial reduction in false connections, suggest that bootstrapping is a valuable strategy for refining the learned causal structures, especially when working with small datasets prone to overfitting and spurious correlations.

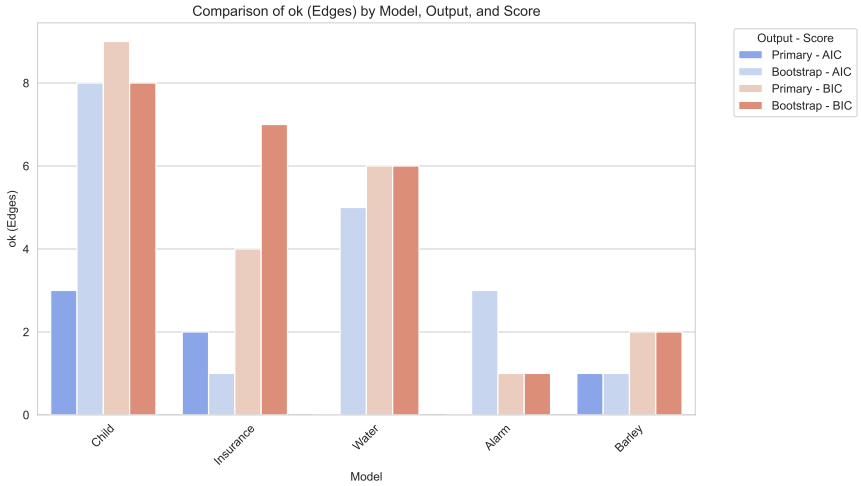
Based on the analysis of the results illustrated in the figures, we observe a consistent pattern indicating that the AIC score outperforms the BIC score across various metrics. Specifically, the AIC score leads to better performance in terms of correctly identified connections ('ok'), fewer missed connections ('miss'), reduced instances of reversed arcs ('rev.'), lower type errors ('type'), and fewer extraneous connections ('xs'). The aggregated 'links' metrics (T, F, M) further corroborate these findings, showing that structures learned using the AIC score are more aligned with the original CBN structures.

The superior performance of the AIC score in our experiments can be attributed to its emphasis on model fit, which proves effective particularly in scenarios where the dataset is sufficiently large to support more complex models without the risk of overfitting. Conversely, the BIC score, which incorporates a stringent penalty for model complexity, tends to be more conservative. This characteristic of BIC can lead to the omission of relevant connections, particularly in smaller datasets where over-penalization can suppress true underlying relationships. Hence, the choice between AIC and BIC should consider the size of the dataset, as it significantly influences their performance and suitability for capturing the true structure of the data.

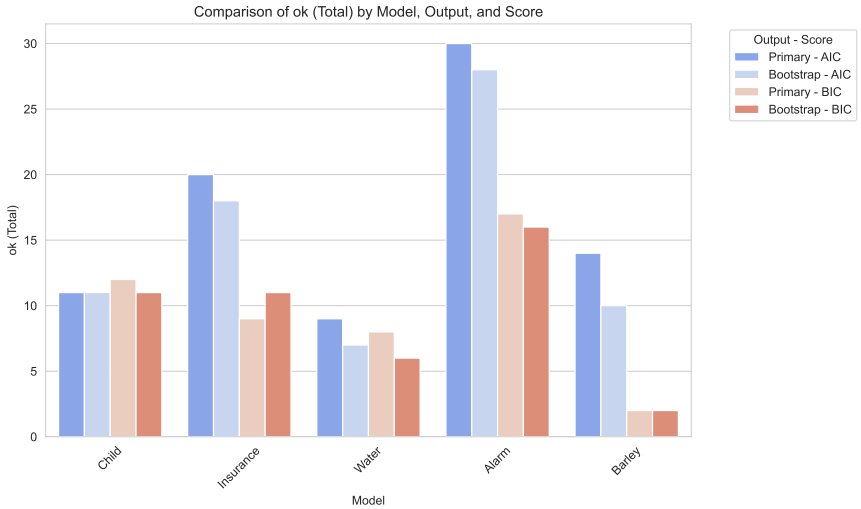
Given the consistent evidence across multiple metrics and models that the AIC score yields structures closer to the original ones, we have decided to utilize the AIC scoring method in the remainder of our experiments. This decision is grounded in the aim to achieve the most accurate and reliable reconstruction of the underlying causal graphs, particularly in the context of medical data, where the precision of the inferred causal relationships is paramount. The enhanced performance of the AIC score in our preliminary experiments provides a strong foundation for this choice, promising more trustworthy and actionable insights in our subsequent analyses.



(a) Arcs

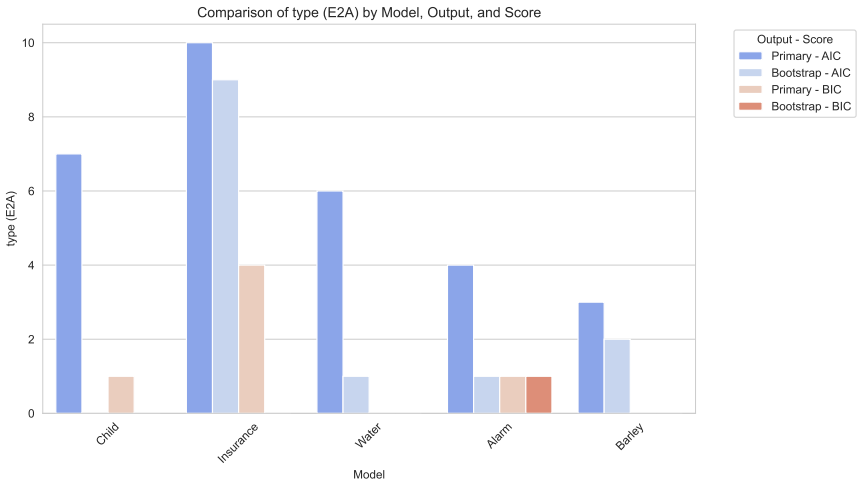


(b) Edges

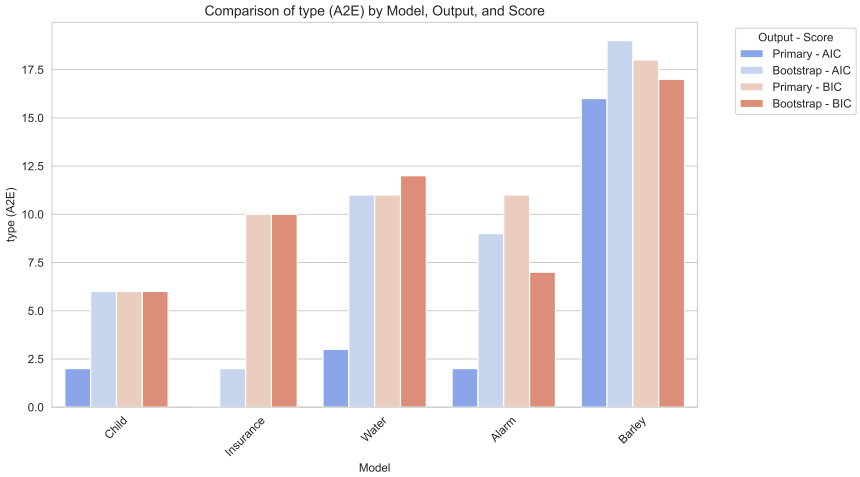


(c) Total

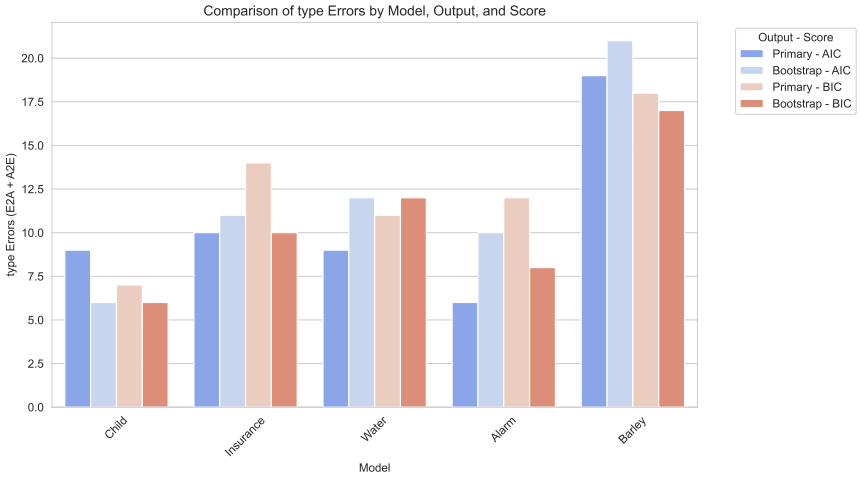
Figure II.16.: Results for 'ok' metric



(a) Arcs

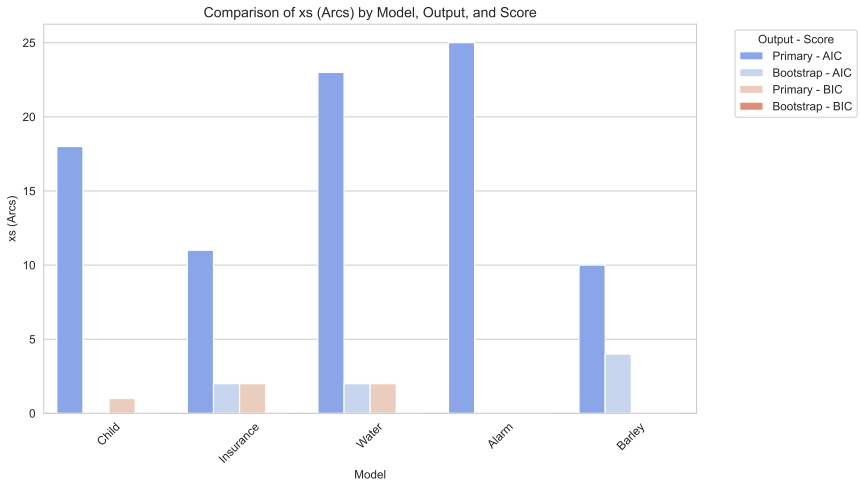


(b) Edges

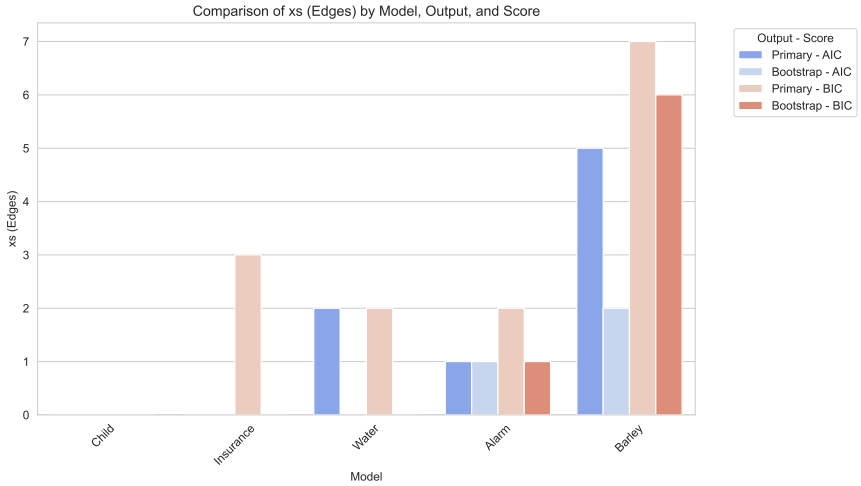


(c) Total

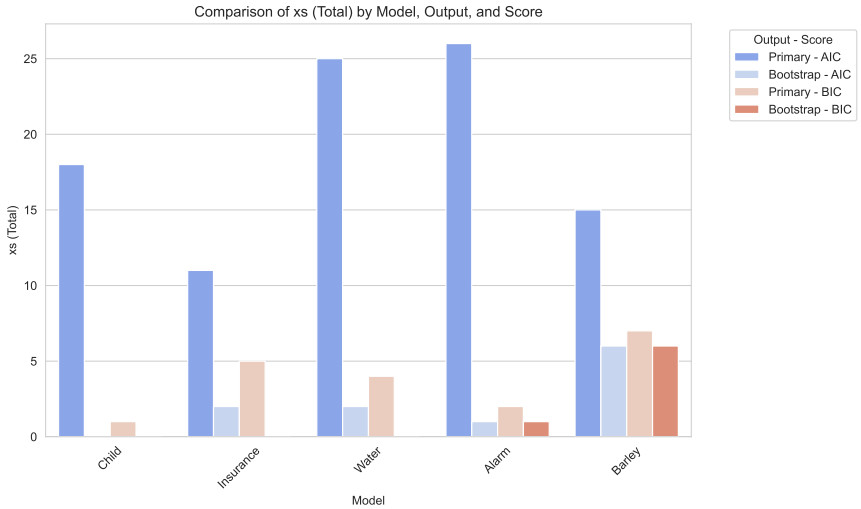
Figure II.17.: Results for 'type' metric



(a) Arcs

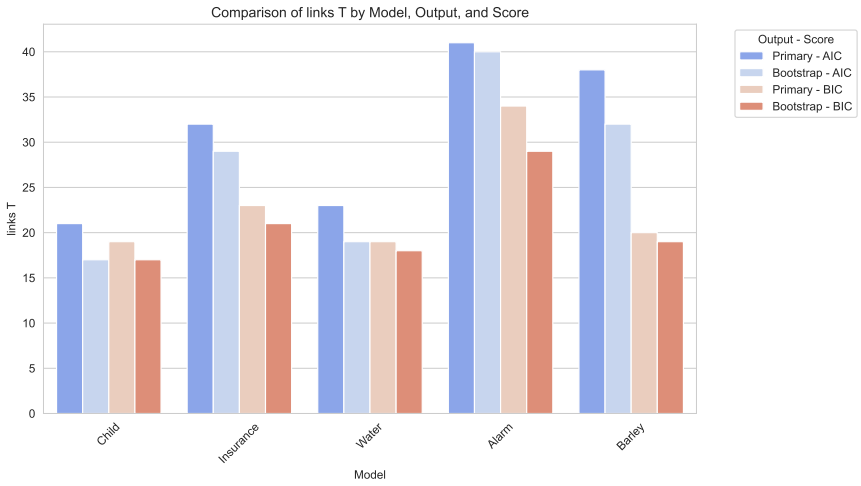


(b) Edges

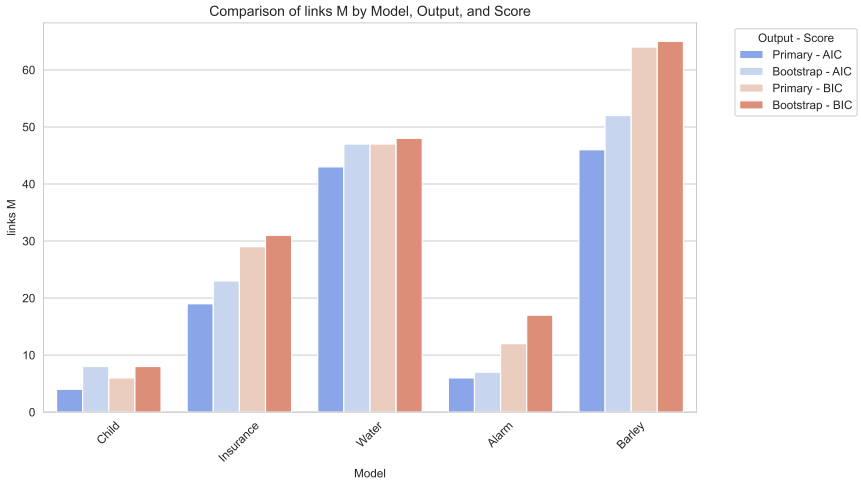


(c) Total

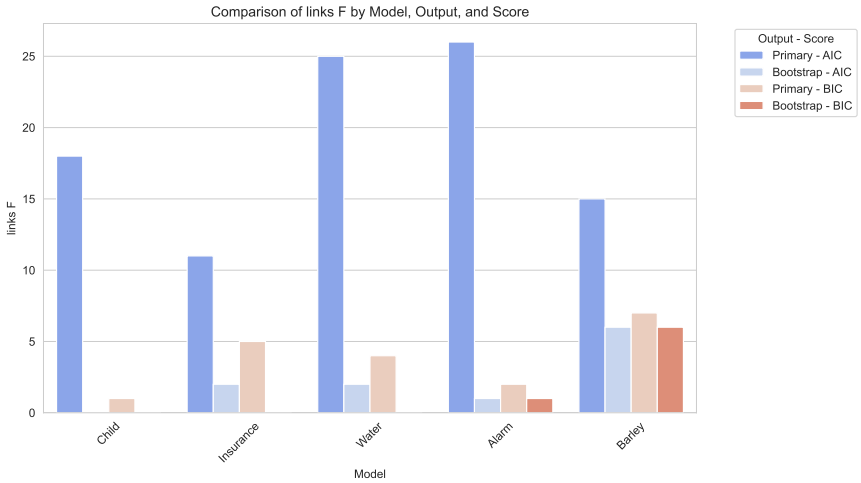
Figure II.18.: Results for 'xs' metric



(a) Arcs



(b) Edges



(c) Total

Figure II.19.: Results for 'links' metric

5.3.3. Bootstrapping and Structure Learning

This section provides a detailed exploration of the experimental procedures applied to the original medical dataset, leveraging our previously validated methodologies for handling small datasets and missing values. The main point of our experiments was the strategic integration of the Expectation Maximization (EM) algorithm and a robust bootstrapping approach, which enriched the dataset and refined the BN model to encapsulate the causal dynamics influencing organ graft surgery outcomes.

As we explained in Algorithm 6, our initial step involved generating bootstrap samples from the original dataset, which contains only 311 samples. This bootstrapping process was critical for mitigating the limitations posed by its small size. We created numerous simulated datasets through resampling with replacement, each serving as a foundation for subsequent BN structure learning. Concurrently, to address the presence of missing data, which was completely random (MCAR), the Expectation-Maximization (EM) algorithm was applied to each bootstrap sample immediately after its generation. This approach not only imputed missing values based on the data's underlying statistical properties and the current network structure but also ensured that each bootstrap sample was complete. This integration of bootstrapping and the EM algorithm facilitated the construction of a more comprehensive and representative dataset, setting the stage for more reliable causal inference.

This iterative generation and refinement of datasets, repeated 1000 times, provided a solid foundation for the subsequent structure learning phase. The resulting collection of complete datasets, each representative of the original data's underlying patterns, enabled us to explore a wide range of potential causal relationships within the medical phenomena under investigation.

For each bootstrap sample, we utilized the CPBayes algorithm to learn DAGs, which were subsequently converted into CPDAGs. This iterative learning process across all bootstrap samples yielded a diverse array of DAGs and CPDAGs, encapsulating potential causal relationships within the data. The aggregation of these structures formed the basis for our subsequent analysis and the derivation of a consensus network model.

5.3.4. Results

Then evaluation follows Algorithm 7, starting with a base CPDAG derived from the original dataset after EM imputation (Algorithm 4). This primary CPDAG, depicted in Figure II.20, is crucial as it helps the subsequent identification of statistically significant causal links through a pre-defined confidence threshold. The ensuing analysis meticulously evaluates these links to construct a reliable and accurate network, reflecting the underlying causal dynamics of the medical phenomena under study.

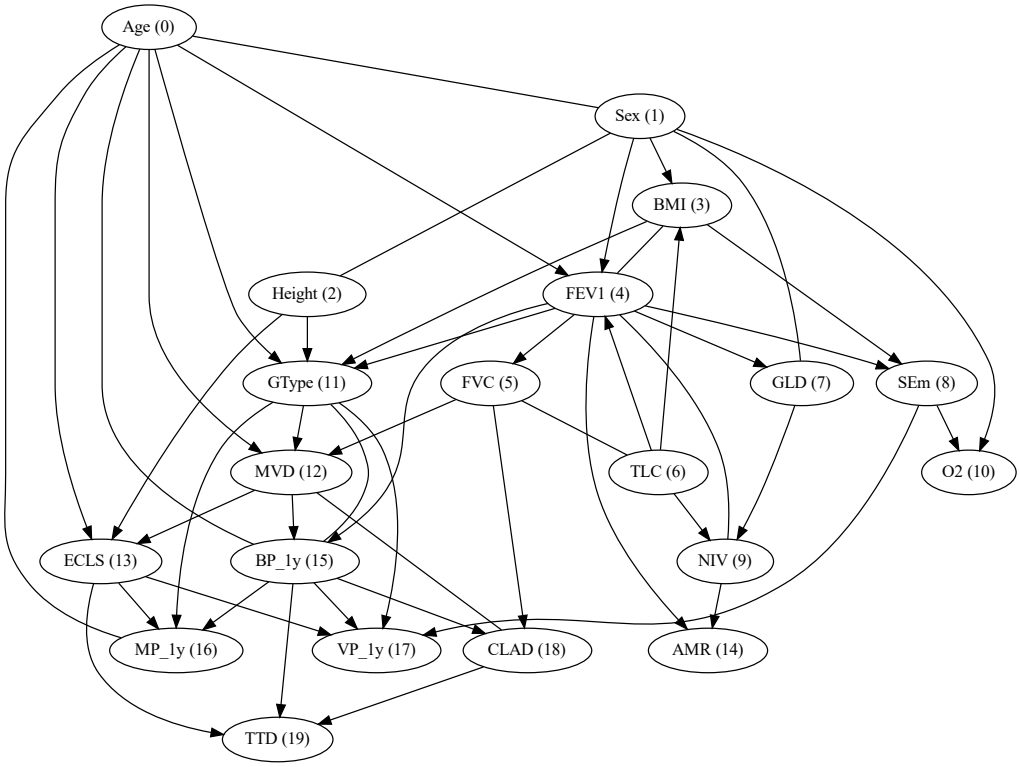


Figure II.20.: Primary learned CPDAG from the imputed original dataset

As discussed earlier, the AIC score was employed for structure learning with the CPBayes algorithm. The resulting primary CPDAG, depicted in Figure II.20, exhibits a high density of arcs. This complexity is indicative of the AIC's inherent preference for more detailed models, as it tends to minimize information loss by selecting models that better fit the data, albeit at the risk of introducing more parameters. The numerous arcs in the CPDAG directly manifest the AIC's tendency to favor complexity, which is aimed at capturing as much of the underlying causal relationships as possible. By applying Algorithm 7, we refine this initial complex network by retaining only those connections that exceed a certain threshold of confidence, ensuring that the final CPDAG comprises only the most reliable and significant arcs. This step effectively reduces the complexity of the model, focusing on the most substantiated causal relationships within the data.

Link Analysis: Our comprehensive analysis of the learned CPDAGs over the bootstrapping process unveiled a rich illustration of potential connections, each evaluated for frequency and consistency across multiple iterations. This rigorous evaluation allowed us to ascertain the confidence levels for each link within the network. Connections that consistently surpassed a predefined confidence threshold were considered statistically significant, meriting inclusion in the final structure. These significant

links form the set $\mathscr{L}_C$ as defined in Algorithm 7, which includes only the most reliable connections. Table II.9 presents a summary of these links, listing each connection alongside its corresponding confidence percentage, irrespective of their orientation.

Table II.9.: Confidence levels for the the learned links ($\mathscr{L}_C$)

Connection	**Confidence Level (%)**
Sex, Height	100.0
FEV1, FVC	100.0
MV_Duration, ECLS	100.0
Age, Graft_Type	99.8
FVC, TLC	98.6
Time_to_Death, BP_1y	94.8
Time_to_Death, ECLS	92.2
Age, Sex	89.5
Graft_Type, BP_1y	87.9
Sex, FEV1	87.4
FEV1, TLC	82.6
Age, FEV1	82.1
Height, Graft_Type	80.6
BMI, SupEmer	77.9
BP_1y, MP_1y	77.1
Age, MV_Duration	75.0
BMI, FEV1	73.8
BMI, TLC	73.2
ECLS, MP_1y	72.0
Sex, Graft_Listing_Duration	70.0
CLAD, BP_1y	68.8
TLC, SupEmer	67.2
Graft_Type, MV_Duration	64.4
Graft_Listing_Duration, NIV	63.3
FEV1, Graft_Type	61.3
Age, BP_1y	60.6
Sex, BMI	60.2

Graph Construction: The next step of our experiment involves transforming the initial complex network depicted in Figure II.20 into a refined graphical representation that underscores the confident causal connections derived from our analysis. By applying Algorithm 7 and utilizing the information from Table II.9, we systematically removed less reliable links and retained only those exceeding our confidence threshold. This process not only streamlined the structure but also ensured that it reflects only the most substantiated causal influences, as depicted in Figure II.21. This graph offers a visual and intuitive portrayal of the critical factors influencing lung transplant

surgery outcomes. The nodes correspond to the variable names and their IDs (in parentheses), which match those in Figure II.20. The weighted arcs within this network indicate the strength of influence exerted by each parent node on their respective child nodes, elucidating the strength and directionality of causal effects within the model as detailed in Section 5.2.5.

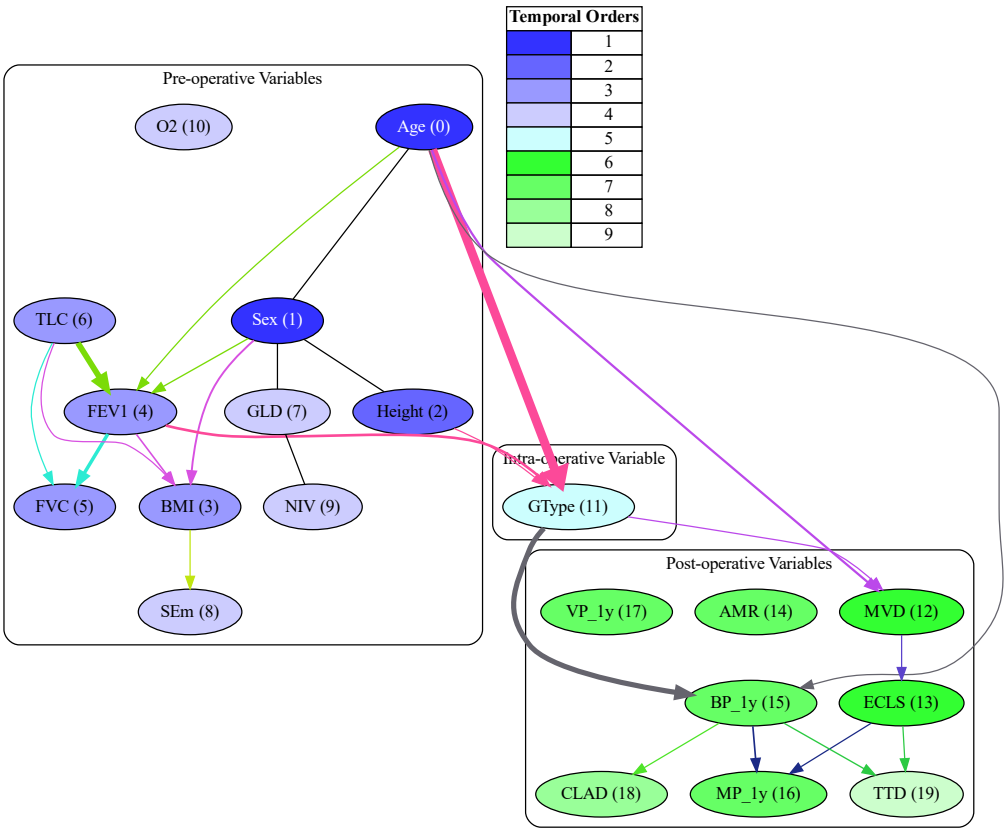


Figure II.21.: Weighted Causal Structure

In this network:

- **Nodes**: Represent the variables in the dataset, with labels corresponding to their names and IDs (colors) based on their orders (as in Table II.5).
- **Directed Arcs**: Indicate causal relationships between variables. The arcs are weighted based on the Shapley values calculated for each parent-child pair, providing a quantitative measure of the influence exerted by parent nodes on their respective child nodes.

- **Arc Thickness**: Corresponds to the normalized Shapley values, visually representing the relative strength of each causal relationship. Thicker arcs denote stronger influence.
- **Undirected Edges**: Represent relationships where the directionality (and thus causality) could not be definitively established. These edges remain unweighted to reflect the ambiguity in their causal interpretation.

Incorporation of Expert Knowledge: In an effort to refine the model's fidelity, we engaged with medical experts to review the temporal order and their knowledge about the causality of some connections within our network. This collaborative process led to the orientation of certain edges based on well-established medical knowledge, thereby enhancing the model's alignment with clinical realities. As previously mentioned in Section 5.2.1, Table II.5 documents the temporal relationships as informed by expert consensus, serving as a guide for the orientation of edges within the network. Based on this Table, the variables from the higher order cannot cause the ones from the lower order. Figure II.22 then presents a revised version of the Causal Network, where selected edges have been directed in accordance with the insights provided by medical experts. This adjustment reinforces the model's credibility and ensures that the depicted causal pathways resonate with established medical understanding.

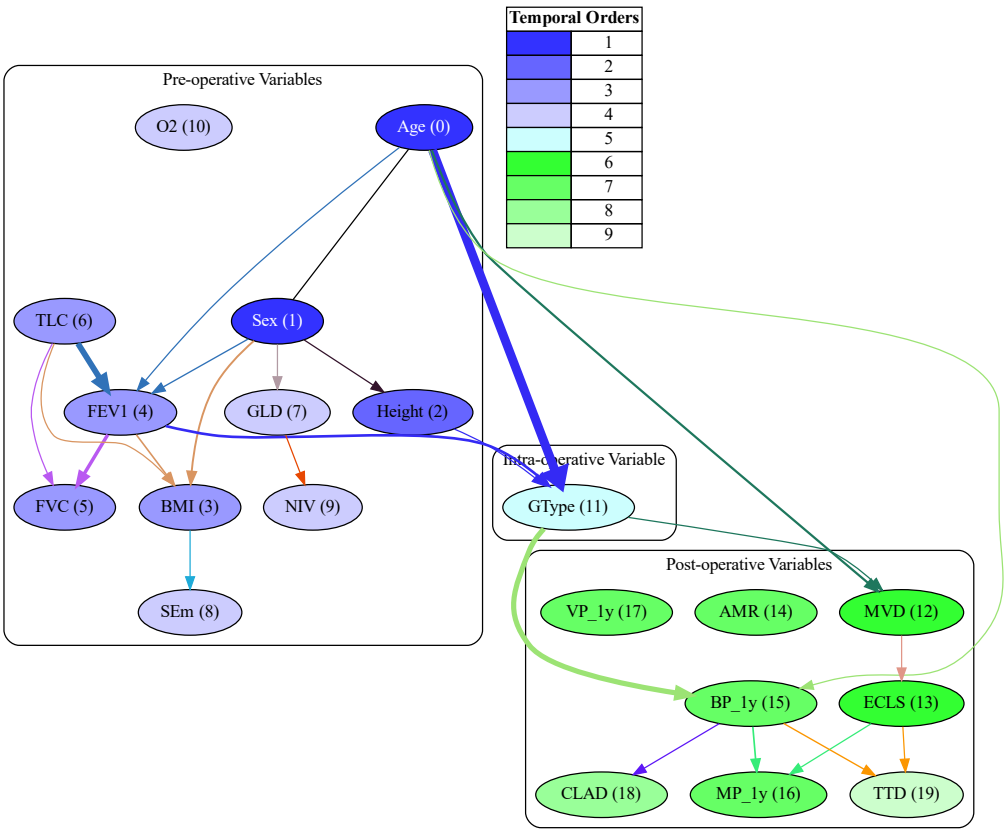


Figure II.22.: Revised Weighted Causal Structure

Latent Variables Identification in Results: Upon analyzing the resultant structures, as depicted in Figure II.22, seven distinct triangles are identified, designated as T_1 to T_7. Specifically, T_1 comprises vertices Age (0), FEV1 (4), and GType (11); T_2 includes Age (0), GType (11), and BP_1y (15); T_3 consists of Age (0), Sex (1), and FEV1 (4); T_4 involves FEV1 (4), FVC (5), and TLC (6); T_5 comprises Age (0), GType (11), and MVD (12); T_6 consists of Sex (1), BMI (3), and FEV1 (4); and T_7 includes BMI (3), and FEV1 (4), and TLC (6). These configurations were systematically evaluated using the approach outlined in Algorithm 2. It was determined that T_2 constitutes a latent triangle of type 1, indicating that latent variables influence these connections. Based on the evaluated scores, the arc between Age (0) and BP_1y (15) is identified as the connection between nodes B and C in a latent triangle, as illustrated in Figure II.7, suggesting its spurious nature due to a latent variable and necessitating its removal as prescribed by Algorithm 2. Consequently, the connection between GType (11) and BP_1y (15) is identified as between nodes A and B, indicating a shared latent parent. Given that BP_1y (15) lacks observable parents, it is inferred that at least one of its children should be considered its parent in the original network structure. Expert

knowledge, as summarized in Table II.5, confirms that Time_to_Death (12) cannot be a parent of BP_1y (15) due to differing temporal orders, leaving MP_1y (18) as the plausible parent. This inference is further supported by dependence tests conducted on the dataset.

However, there was insufficient evidence to classify T_1, T_3 to T_7 as latent triangles. Limited dataset size constrained our ability to detect additional latent variables reliably in these triangles. With a larger dataset, it might have been possible to ascertain the presence of latent variables in T_1, T_3 to T_7 as well. Following this analysis, modifications were applied to the network structure to remove spurious connections and incorporate the identified latent variables, leading to an updated and more accurate representation of the causal dynamics within the data, as illustrated in Figure II.23.

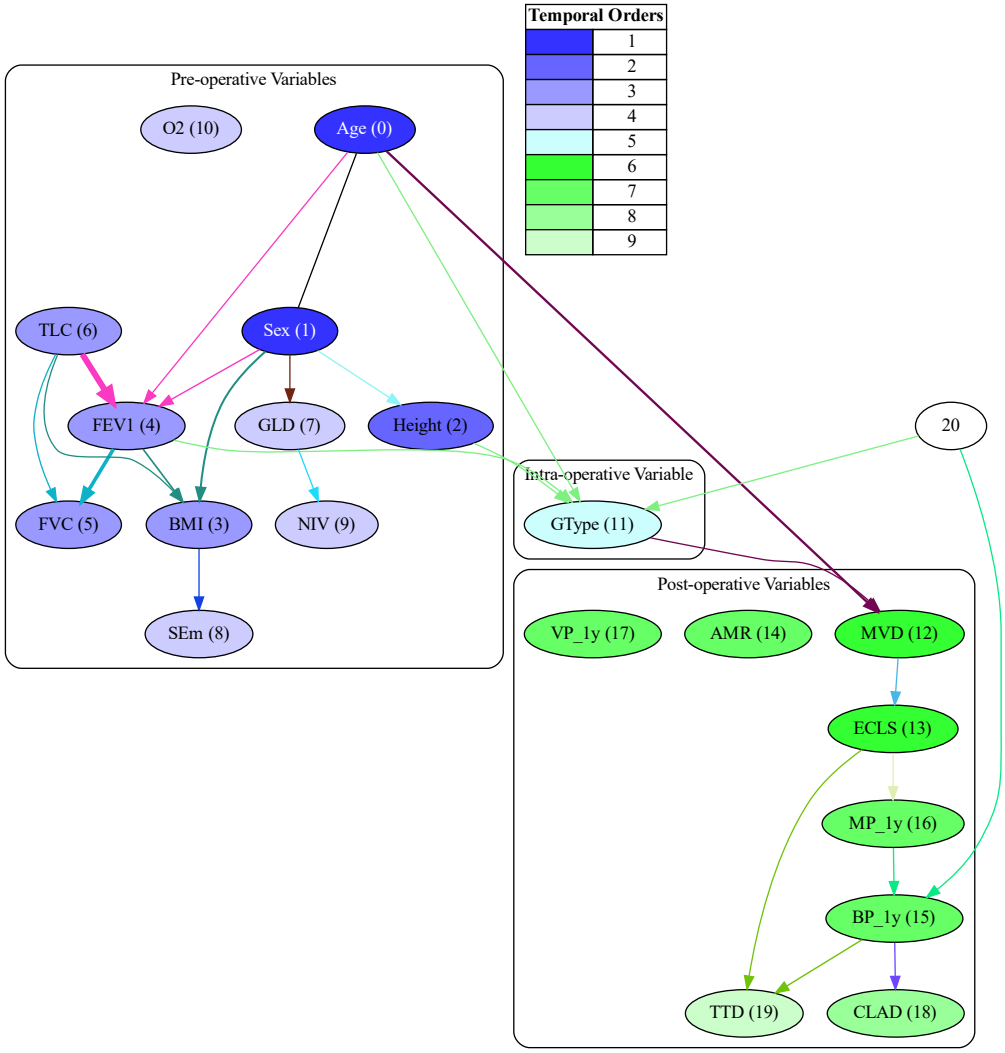


Figure II.23.: Final Causal Structure with Latent Variable

Expert Validation of Causal Relationships: Medical experts thoroughly reviewed and validated the causal links identified through our methodology. They confirmed that the causal relationships reflected in our final network model align with established clinical knowledge and insights into lung graft surgery outcomes. This validation underscores our findings' practical relevance and reliability and endorses the methodological rigor applied in this study.

5.3.5. Discussion

After conducting our experiments, we are able to express the implications of our findings and the methodologies employed in our causal learning approach for medical data, particularly focusing on organ graft surgery outcomes. The integration of bootstrapping techniques with Expectation Maximization (EM) algorithms for handling missing data has proven instrumental in augmenting the dataset and ensuring its completeness. Furthermore, our innovative approach to identifying latent variables within the dataset has significantly refined the construction of a trustworthy Bayesian Network model, capturing the intricate causal relationships within the medical data.

The construction of a reliable network structure, underscored by high-confidence arcs and edges, reflects the robustness of our methodological framework and offers significant insights into the variables influencing lung graft surgery outcomes. The validation of the bootstrapping technique, coupled with the iterative learning of DAGs and their refinement through latent variables, has been crucial in affirming the stability and reliability of the inferred causal relationships.

Furthermore, the incorporation of domain expertise, particularly in the orientation of non-causal links within the CPDAG, has been invaluable. This collaborative approach has ensured that the final model is not only data-driven but also aligns with established medical knowledge, thereby enhancing its clinical relevance and applicability.

In addition to our analytical evaluations, medical experts have extensively validated the causal relationships and network structures derived from our study. Their expertise in the clinical aspects of lung transplant outcomes has further affirmed the accuracy of our findings. This collaborative validation process ensures that the model not only adheres to statistical rigor but also resonates with practical, real-world medical insights. The expert endorsement of the causal links enhances the credibility and applicability of our research, providing a robust foundation for future clinical applications and studies.

5.4. Conclusion

Our research presents a comprehensive causal learning approach tailored for medical data, with a focus on lung graft surgery outcomes. Through the rigorous application of bootstrapping and EM algorithms, coupled with a novel method for latent variable identification, we have successfully addressed the challenges of small dataset sizes and missing data, laying a solid foundation for the subsequent construction of a Bayesian Network model.

The experiments conducted have produced a robust model that not only captures the complex causal dynamics within the medical data but also has been thoroughly validated by medical experts. This validation has reinforced the model's validity, aligning it closely with clinical realities and ensuring its relevance for practical decision-making

in healthcare settings. Thus, the model serves as a significant step forward in understanding the factors influencing lung graft surgery outcomes, offering valuable insights for clinical decision-making and future research.

The methodologies and findings from this study underscore the potential of causal learning approaches in medical research, especially in scenarios constrained by data limitations. As we move forward, it is imperative to continue refining these methodologies and exploring their application across various medical domains to uncover deeper causal insights that can inform clinical practices and improve patient outcomes.

III. Conclusion and Future Works

Sommaire

6. Conclusion 151
7. Future Works 152

6. Conclusion

This thesis undertook an exploration into the complex field of causal learning with Bayesian Networks (BNs), with an initial focus on understanding the challenges and opportunities associated with latent variables. The investigation included a comprehensive review of recent advancements in causal networks, learning algorithms, and the critical role that latent variables play in defining the causal structure of BNs.

In Part I, we delved into the foundational aspects of Bayesian and Causal Networks, setting the stage for a deeper inquiry into causal learning. We distinguished between observational and interventional data, highlighting the critical need for methodologies capable of discerning causal relationships from mere correlations. The exploration of causal models, particularly those incorporating latent variables, revealed the complexities inherent in inferring causality in the presence of unobserved factors.

Part II represents a central part of this thesis, where we introduce a score-based causal learning algorithm designed to improve the handling of latent variables, contributing to the ongoing efforts to unveil underlying causal structures from data. This algorithm seeks to enhance the management of latent variables, thereby enriching our understanding of causal structures embedded within complex datasets. Employing triangle motifs in Directed Acyclic Graphs (DAGs), the algorithm aims to provide a robust framework for identifying latent variables and clarifying their roles within the causal network.

During the experiments detailed in the thesis, we evaluated the effectiveness of the proposed algorithm, which showed promising potential. By applying the algorithm to classical benchmark Causal Bayesian Networks (CBNs), we were able to distinguish genuine from latent triangles, revealing deeper layers of causality that were obscured by latent variables. In comparison with existing algorithms such as MIIC and FCI, the results suggest that our algorithm offers improvements in precision, recall, and the F1

score, particularly as the dataset size increases. These findings indicate a potential for enhancing the robustness of causal inference under certain conditions.

As the research progressed, it became apparent that the challenges associated with small datasets and missing data in medical research required a more refined approach. Consequently, the latter part of this thesis focused on developing and validating a strategy tailored specifically for medical data. This approach, which integrates latent variable identification with bootstrapping and EM algorithm, is designed to enhance the construction of causal Bayesian networks. It aims to facilitate a deeper understanding of complex medical conditions and treatment outcomes, contributing to the ongoing efforts in medical research.

The implications of this work are promising and may extend beyond academia, suggesting potential applications in diverse fields such as healthcare, epidemiology, economics, and social sciences. The nuanced understanding of causal structures gained, particularly in medical research, could open new avenues for interventions, policy formulation, and innovative strategies. However, these are initial insights and further research is necessary to fully realize and apply these findings effectively in practical settings.

In conclusion, this thesis offers a fresh perspective and a set of tools to the domain of causal learning, establishing a groundwork for future explorations. While challenges persist, particularly with latent variables in causal inference, this research contributes to ongoing efforts in developing more sophisticated models and algorithms. As the field continues to explore the complexities of causality, the insights provided by this thesis may prove useful in enhancing our collective understanding of various phenomena. The potential impact of these contributions will be better understood through continued research and practical application.

7. Future Works

Building upon the approaches introduced in this thesis, future research holds promising possibilities for refining and extending the capabilities of causal learning algorithms. The development and validation of a score-based causal learning algorithm, alongside a comprehensive strategy to address missing data and small dataset sizes, have established a foundation for further exploration. Looking ahead, we plan to enhance computational efficiency by potentially substituting the CPBayes with a faster, approximate algorithm such as Greedy Hill Climbing (GHC) in Line 1 of Algorithm2. While GHC-like methods may increase efficiency, they often introduce more errors in the directions of causal arcs compared to more precise methods like CPBayes. This challenge will necessitate the refinement of the rules used in Line 5-9 of the algorithm to mitigate potential inaccuracies.

Moreover, while our current experimental setup seldom encountered Type2 triangles, leading to their exclusion from significant consideration in Algorithm2, the

shift to GHC-like methods may increase their prevalence. This change underscores the need for a more inclusive framework capable of adeptly handling various triangle configurations, ensuring no latent confounder is overlooked due to the limitations of the methodologies used.

Furthermore, a more immediate opportunity for improvement lies in refining algorithms to correct directional biases observed in specific triangle configurations. An example of such bias is found in Type3 triangles, where the connection from $A \longrightarrow C$ often appears in the incorrect direction (see Figure II.6b). To enhance the accuracy and reliability of the inferred CPDAGs, it is proposed that Algorithm2 should reverse this arc and subsequently relearn the neighborhood of node C. By systematically addressing these misdirections and re-evaluating the neighborhoods of affected nodes, we can significantly improve the quality of the causal models produced.

It is worth exploring the potential benefits of combining constraint-based methods, such as FCI, with the developed algorithm to unlock new synergies. FCI is known for its proficiency in identifying the majority of latent confounders, which may correspond to triangles that our algorithm has incorrectly identified. By integrating these methods, we could leverage complementary strengths, potentially enhancing the accuracy and efficiency of Algorithm 2. This joint approach may offer a promising direction for methodological advancements, improving instances where triangles are misclassified and thus refining the overall performance of our causal learning process.

The refined causal learning framework we have developed has the potential to be applied across many different fields as what we did for the medical domain. By adapting the algorithm to specific domain challenges and utilizing its improved capabilities, there is an opportunity to gain deeper insights into complex phenomena in areas like healthcare, economics, and social sciences. These interdisciplinary applications could help demonstrate the algorithm's versatility and contribute to a more nuanced understanding of causality in various contexts.

In essence, the future of causal learning presents many opportunities for continued innovation, interdisciplinary application, and methodological refinement. The explorations outlined in this thesis provide a foundation that encourages further investigation into the intricate nature of causality that governs our world.

Possible future directions can be summarized as follows:

- **Scalability and Efficiency:** Exploration of faster approximate algorithms like Greedy Hill Climbing (GHC) to enhance the scalability and computational efficiency of causal learning algorithms.
- **Methodological Refinement:** Development of advanced rules and strategies to accommodate the directional inaccuracies introduced by approximate algorithms and ensure comprehensive triangle configuration handling.

- **Directional Bias Correction:** Implementation of corrective measures for misdirected connections in (Type 3) triangles to improve the accuracy of inferred CPDAGs.
- **Hybrid Approaches:** Integration of constraint-based insights with score-based methodologies to refine triangle classification and confounder identification processes.
- **Interdisciplinary Applications:** Tailoring the refined causal learning framework to domain-specific challenges to provide deeper causal insights across various fields.

These directions aim to contribute to advancing the field of causal learning and may open new avenues for impactful research and application. Such explorations could further our understanding of the causal mechanisms that shape complex systems across disciplines.

Liste de participation aux conférences

Participation aux conférences au cours de la période de thèse:

1. **11èmes Journées Francophones sur les Réseaux Bayésiens et les Modèles Graphiques Probabilistes (JFRB 2023)**, 1-2 juin 2023 Nantes, LS2N, site de Polytech'Nantes, France *"Un algorithme efficace fondé sur les scores pour l'apprentissage de structure de réseaux causaux avec variables latentes"* Christophe GONZALES, Amir Hosein VALIZADEH
2. **27th European Conference on Artificial Intelligence (ECAI 2024)**, 19-24 October 2024, Santiago de Compostela, Spain *"A Full DAG Score-Based Algorithm for Learning Causal Bayesian Networks with Latent Confounders"* Christophe GONZALES, Amir Hosein VALIZADEH

Bibliography

[Aka70] Hirotugu Akaike. “Statistical predictor identification”. In: *Annals of the Institute of Statistical Mathematics* 22.1 (1970), pp. 203–217 (cit. on pp. 70, 71).

[ALW21] Mahdi Hadj Ali, Yann Le Biannic, and Pierre-Henri Wuillemin. “Réseaux bayésiens et valeurs de Shapley”. In: *10èmes Journée Francophones sur les Réseaux Bayésiens et les Modèles Graphiques Probabilistes (JFRB 2021)*. 2021 (cit. on p. 127).

[ARS04] R Ayesha Ali, Thomas Richardson, and Peter Spirtes. *Markov equivalence for ancestral graphs*. Tech. rep. Technical Report 466, Department of Statistics, University of Washington, 2004 (cit. on pp. 45, 48).

[Ali+10] Constantin F Aliferis, Alexander Statnikov, Ioannis Tsamardinos, et al. “Local causal and Markov blanket induction for causal discovery and feature selection for classification part I: algorithms and empirical evaluation.” In: *Journal of Machine Learning Research* 11.1 (2010) (cit. on p. 62).

[ATS03] Constantin F Aliferis, Ioannis Tsamardinos, and Alexander Statnikov. “HITON: a novel Markov Blanket algorithm for optimal variable selection”. In: *AMIA annual symposium proceedings*. Vol. 2003. American Medical Informatics Association. 2003, p. 21 (cit. on p. 62).

[Ali+03] Constantin F Aliferis, Ioannis Tsamardinos, Alexander R Statnikov, et al. “Causal Explorer: A Causal Probabilistic Network Learning Toolkit for Biomedical Discovery.” In: *METMBS*. Vol. 3. 2003, pp. 371–376 (cit. on p. 62).

[AMP97] Steen A Andersson, David Madigan, and Michael D Perlman. “A characterization of Markov equivalence classes for acyclic digraphs”. In: *The Annals of Statistics* 25.2 (1997), pp. 505–541 (cit. on p. 34).

[Bar+22] Elias Bareinboim, Juan D Correa, Duligur Ibeling, et al. “On Pearl’s hierarchy and the foundations of causal inference”. In: *Probabilistic and causal inference: the works of judea pearl*. 2022, pp. 507–556 (cit. on pp. 28, 35).

[Bil17] Patrick Billingsley. *Probability and measure*. John Wiley & Sons, 2017 (cit. on p. 20).

[Bou93] Remco R Bouckaert. "Probabilistic network construction using the minimum description length principle". In: *European conference on symbolic and quantitative approaches to reasoning and uncertainty*. Springer. 1993, pp. 41–48 (cit. on p. 97).

[Bou94] Remco R Bouckaert. "Properties of Bayesian belief network learning algorithms". In: *Uncertainty Proceedings 1994*. Elsevier, 1994, pp. 102–109 (cit. on pp. 51, 73).

[Bou95] Remco Ronaldus Bouckaert. "Bayesian belief networks: from construction to inference". PhD thesis. 1995 (cit. on p. 73).

[Bun91] Wray Buntine. "Theory refinement on Bayesian networks". In: *Uncertainty proceedings 1991*. Elsevier, 1991, pp. 52–60 (cit. on pp. 67, 69).

[Cam+18] Cassio P de Campos, Mauro Scanagatta, Giorgio Corani, et al. "Entropy-based pruning for learning Bayesian networks using BIC". In: *Artificial Intelligence* 260 (2018), pp. 42–50 (cit. on p. 76).

[Chi95] David Maxwell Chickering. "A transformational characterization of equivalent Bayesian network structures". In: *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*. 1995, pp. 87–98 (cit. on p. 104).

[Chi96] David Maxwell Chickering. "Learning Bayesian networks is NP-complete". In: *Learning from data: Artificial intelligence and statistics V* (1996), pp. 121–130 (cit. on p. 57).

[Chi02a] David Maxwell Chickering. "Learning equivalence classes of Bayesian-network structures". In: *The Journal of Machine Learning Research* 2 (2002), pp. 445–498 (cit. on pp. 32, 74, 75).

[Chi02b] David Maxwell Chickering. "Optimal structure identification with greedy search". In: *Journal of machine learning research* 3.Nov (2002), pp. 507–554 (cit. on p. 74).

[CC20] Kiattikun Chobtham and Anthony C Constantinou. "Bayesian network structure learning with causal effects in the presence of latent variables". In: *International Conference on Probabilistic Graphical Models*. PMLR. 2020, pp. 101–112 (cit. on p. 78).

[CMH13] Tom Claassen, Joris Mooij, and Tom Heskes. "Learning sparse causal models is not NP-hard". In: *arXiv preprint arXiv:1309.6824* (2013) (cit. on p. 64).

[CM+14] Diego Colombo, Marloes H Maathuis, et al. "Order-independent constraint-based causal structure learning." In: *J. Mach. Learn. Res.* 15.1 (2014), pp. 3741–3782 (cit. on pp. 61, 64).

[Col+12] Diego Colombo, Marloes H Maathuis, Markus Kalisch, et al. "Learning high-dimensional directed acyclic graphs with latent and selection variables". In: *The Annals of Statistics* (2012), pp. 294–321 (cit. on p. 64).

[Con+22] Anthony C Constantinou, Yang Liu, Neville K Kitson, et al. "Effective and efficient structure learning with pruning and model averaging strategies". In: *International Journal of Approximate Reasoning* 151 (2022), pp. 292–321 (cit. on p. 73).

[CH92] Gregory F Cooper and Edward Herskovits. "A Bayesian method for the induction of probabilistic networks from data". In: *Machine learning* 9 (1992), pp. 309–347 (cit. on pp. 51, 67, 68, 73).

[CCC20] Alvaro Henrique Chaim Correia, James Cussens, and Cassio de Campos. "On pruning for score-based Bayesian network structure learning". In: *International conference on artificial intelligence and statistics*. PMLR. 2020, pp. 2709–2718 (cit. on p. 76).

[Cus12] James Cussens. "An upper bound for BDeu local scores". In: *Submitted, May* (2012) (cit. on p. 75).

[DJ11] Cassio P De Campos and Qiang Ji. "Efficient structure learning of Bayesian networks using constraints". In: *The Journal of Machine Learning Research* 12 (2011), pp. 663–689 (cit. on p. 76).

[DZJ09] Cassio P De Campos, Zhi Zeng, and Qiang Ji. "Structure learning of Bayesian networks using constraints". In: *Proceedings of the 26th Annual International Conference on Machine Learning*. 2009, pp. 113–120 (cit. on pp. 51, 75, 76).

[DLR77] Arthur P Dempster, Nan M Laird, and Donald B Rubin. "Maximum likelihood from incomplete data via the EM algorithm". In: *Journal of the royal statistical society: series B (methodological)* 39.1 (1977), pp. 1–22 (cit. on p. 120).

[DGW20] Gaspard Ducamp, Christophe Gonzales, and Pierre-Henri Wuillemin. "aGrUM/pyAgrum: a toolbox to build models and algorithms for Probabilistic Graphical Models in Python". In: *International Conference on Probabilistic Graphical Models*. PMLR. 2020 (cit. on pp. 107, 121, 133).

[Fri98] Nir Friedman. "The Bayesian structural EM algorithm". In: *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*. 1998, pp. 129–138 (cit. on pp. 120, 122).

[FGW13] Nir Friedman, Moises Goldszmidt, and Abraham Wyner. "Data analysis with Bayesian networks: A bootstrap approach". In: *arXiv preprint arXiv:1301.6695* (2013) (cit. on p. 123).

[FK03] Nir Friedman and Daphne Koller. "Being Bayesian about network structure. A Bayesian approach to structure discovery in Bayesian networks". In: *Machine learning* 50 (2003), pp. 95–125 (cit. on p. 78).

[FNP99] Nir Friedman, Iftach Nachman, and Dana Peér. "Learning bayesian network structure from massive datasets: the sparse candidate algorithm". In: *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*. 1999, pp. 206–215 (cit. on p. 77).

[GP98] David Galles and Judea Pearl. "An axiomatic characterization of causal counterfactuals". In: *Foundations of Science* 3 (1998), pp. 151–182 (cit. on p. 36).

[GP90] Dan Geiger and Judea Pearl. "On the logic of causal models". In: *Machine Intelligence and Pattern Recognition*. Vol. 9. Elsevier, 1990, pp. 3–14 (cit. on pp. 22–24).

[Glo90] Fred Glover. "Tabu search—part II". In: *ORSA Journal on computing* 2.1 (1990), pp. 4–32 (cit. on pp. 51, 73).

[GZS19] Clark Glymour, Kun Zhang, and Peter Spirtes. "Review of causal discovery methods based on graphical models". In: *Frontiers in genetics* 10 (2019), p. 524 (cit. on p. 49).

[GV24] Christophe Gonzales and Amir-Hosein Valizadeh. "A Full DAG Score-Based Algorithm for Learning Causal Bayesian Networks with Latent Confounders". In: *ECAI 2024*. IOS Press, 2024, pp. 1712–1719 (cit. on p. 113).

[Hal00] Joseph Y Halpern. "Axiomatizing causal reasoning". In: *Journal of Artificial Intelligence Research* 12 (2000), pp. 317–337 (cit. on p. 36).

[HGC95] David Heckerman, Dan Geiger, and David M Chickering. "Learning Bayesian networks: The combination of knowledge and statistical data". In: *Machine learning* 20 (1995), pp. 197–243 (cit. on pp. 52, 53, 67–69, 73).

[Hol86] Paul W Holland. "Statistics and causal inference". In: *Journal of the American statistical Association* 81.396 (1986), pp. 945–960 (cit. on pp. 40, 41).

[Jef46] Harold Jeffreys. "An invariant form for the prior probability in estimation problems". In: *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* 186.1007 (1946), pp. 453–461 (cit. on p. 68).

[Kit+23] Neville Kenneth Kitson, Anthony C Constantinou, Zhigao Guo, et al. "A survey of Bayesian Network structure learning". In: *Artificial Intelligence Review* (2023), pp. 1–94 (cit. on p. 76).

[KSB17] Murat Kocaoglu, Karthikeyan Shanmugam, and Elias Bareinboim. "Experimental design for learning causal graphs with latent variables". In: *Advances in Neural Information Processing Systems* 30 (2017) (cit. on p. 44).

[KS04] Mikko Koivisto and Kismat Sood. "Exact Bayesian structure discovery in Bayesian networks". In: *The Journal of Machine Learning Research* 5 (2004), pp. 549–573 (cit. on pp. 51, 75).

[KF09] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009 (cit. on pp. 17, 27, 30, 31, 37, 38, 50, 53, 67, 84, 97).

[KB18] Andre Nikolaevich Kolmogorov and Albert T Bharucha-Reid. *Foundations of the theory of probability: Second English Edition*. Courier Dover Publications, 2018 (cit. on p. 19).

[KM07] Petri Kontkanen and Petri Myllymäki. "A linear-time algorithm for computing the multinomial stochastic complexity". In: *Information Processing Letters* 103.6 (2007), pp. 227–233 (cit. on p. 72).

[KM17] Jack Kuipers and Giusi Moffa. "Partition MCMC for inference on acyclic digraphs". In: *Journal of the American Statistical Association* 112.517 (2017), pp. 282–299 (cit. on p. 78).

[KSM22] Jack Kuipers, Polina Suter, and Giusi Moffa. "Efficient sampling and structure learning of Bayesian networks". In: *Journal of Computational and Graphical Statistics* 31.3 (2022), pp. 639–650 (cit. on p. 78).

[LB94] Wai Lam and Fahiem Bacchus. "Learning Bayesian belief networks: An approach based on the MDL principle". In: *Computational intelligence* 10.3 (1994), pp. 269–293 (cit. on pp. 70, 71).

[LMY12] Zhifa Liu, Brandon Malone, and Changhe Yuan. "Empirical evaluation of scoring functions for Bayesian network model selection". In: *BMC bioinformatics*. Vol. 13. 15. BioMed Central. 2012, pp. 1–16 (cit. on pp. 71, 72).

[Mar+03] Dimitris Margaritis et al. "Learning Bayesian network model structure from data". PhD thesis. School of Computer Science, Carnegie Mellon University Pittsburgh, PA, USA, 2003 (cit. on p. 62).

[MT99] Dimitris Margaritis and Sebastian Thrun. "Bayesian network induction via local neighborhoods". In: *Advances in neural information processing systems* 12 (1999) (cit. on p. 62).

[MGS05] Florian Markowetz, Steffen Grossmann, and Rainer Spang. "Probabilistic soft interventions in conditional Gaussian networks". In: *International Workshop on Artificial Intelligence and Statistics*. PMLR. 2005, pp. 214–221 (cit. on p. 36).

[MV95] J Martin and Kurt VanLehn. *Discrete factor analysis: Learning hidden variables in Bayesian networks*. Tech. rep. Technical report, Department of Computer Science, University of Pittsburgh, 1995 (cit. on p. 30).

[Mee95] Christopher Meek. "Causal Inference and Causal Explanation with Background Knowledge". In: UAI'95 (1995), pp. 403–410 (cit. on pp. 34, 58, 60, 74, 102).

[MC20] Joris M Mooij and Tom Claassen. "Constraint-based causal discovery using partial ancestral graphs in the presence of cycles". In: *Conference on Uncertainty in Artificial Intelligence*. PMLR. 2020, pp. 1159–1168 (cit. on p. 33).

[MMC20] Joris M Mooij, Sara Magliacane, and Tom Claassen. "Joint causal inference from multiple contexts". In: *The Journal of Machine Learning Research* 21.1 (2020), pp. 3919–4026 (cit. on p. 33).

[Nea20] Brady Neal. "Introduction to causal inference". In: *Course Lecture Notes (draft)* (2020) (cit. on pp. 40, 43).

[NLR11] Hoai-Tuong Nguyen, Philippe Leray, and Gérard Ramstein. "Summarizing and visualizing a set of bayesian networks with quasi essential graphs". In: *ASMDA 2011*. 2011, pp. 1062–1069 (cit. on p. 124).

[Nog+22] Ana Rita Nogueira, Andrea Pugnana, Salvatore Ruggieri, et al. "Methods and tools for causal discovery and causal inference". In: *Wiley interdisciplinary reviews: data mining and knowledge discovery* 12.2 (2022), e1449 (cit. on p. 43).

[OSR16] Juan Miguel Ogarrio, Peter Spirtes, and Joe Ramsey. "A hybrid causal search algorithm for latent variable models". In: *Conference on probabilistic graphical models*. PMLR. 2016, pp. 368–379 (cit. on p. 50).

[OIM03] Sascha Ott, Seiya Imoto, and Satoru Miyano. "Finding optimal models for small gene networks". In: *Biocomputing 2004*. World Scientific, 2003, pp. 557–567 (cit. on pp. 51, 75).

[Pea88] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan kaufmann, 1988 (cit. on pp. 22, 24, 32, 62, 98).

[Pea93] Judea Pearl. "[Bayesian analysis in expert systems]: comment: graphical models, causality and intervention". In: *Statistical Science* 8.3 (1993), pp. 266–269 (cit. on p. 37).

[Pea94] Judea Pearl. "A probabilistic calculus of actions". In: *Uncertainty Proceedings 1994*. Elsevier, 1994, pp. 454–462 (cit. on pp. 35, 36).

[Pea95] Judea Pearl. "Causal diagrams for empirical research". In: *Biometrika* 82.4 (1995), pp. 669–688 (cit. on p. 38).

[Pea09] Judea Pearl. *Causality*. Cambridge university press, 2009 (cit. on pp. 29, 30, 32, 35–38, 50).

[PM18] Judea Pearl and Dana Mackenzie. *The book of why: the new science of cause and effect*. Basic books, 2018 (cit. on p. 35).

[PIM08] Eric Perrier, Seiya Imoto, and Satoru Miyano. "Finding Optimal Bayesian Network Given a Super-Structure." In: *Journal of Machine Learning Research* 9.10 (2008) (cit. on p. 77).

[PS88] John W Pratt and Robert Schlaifer. "On the interpretation and observation of laws". In: *Journal of Econometrics* 39.1-2 (1988), pp. 23–52 (cit. on p. 42).

[Ram16] Joseph Ramsey. "Improving accuracy and scalability of the pc algorithm by maximizing p-value". In: *arXiv preprint arXiv:1610.00378* (2016) (cit. on p. 61).

[Ram+17] Joseph Ramsey, Madelyn Glymour, Ruben Sanchez-Romero, et al. "A million variables and more: the fast greedy equivalence search algorithm for learning high-dimensional graphical causal models, with an application to functional magnetic resonance images". In: *International journal of data science and analytics* 3 (2017), pp. 121–129 (cit. on pp. 74, 76).

[RZS06] Joseph Ramsey, Jiji Zhang, and Peter Spirtes. "Adjacency-Faithfulness and Conservative Causal Inference". In: *Proceedings of the Twenty-Second Conference Conference on Uncertainty in Artificial Intelligence (2006)*. Ed. by R. Dechter and T. Richardson. AUAI Press, 2006, pp. 401–408 (cit. on pp. 61, 64).

[RS99] Thomas Richardson and Peter Spirtes. "Scoring ancestral graph models". In: (1999) (cit. on p. 50).

[RS02] Thomas Richardson and Peter Spirtes. "Ancestral graph Markov models". In: *The Annals of Statistics* 30.4 (2002), pp. 962–1030 (cit. on pp. 45, 46).

[Ric09] Thomas S Richardson. "A factorization criterion for acyclic directed mixed graphs". In: *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*. 2009, pp. 462–470 (cit. on p. 75).

[Ris96] Jorma J Rissanen. "Fisher information and stochastic complexity". In: *IEEE transactions on information theory* 42.1 (1996), pp. 40–47 (cit. on pp. 70, 71).

[RR83] Paul R Rosenbaum and Donald B Rubin. "The central role of the propensity score in observational studies for causal effects". In: *Biometrika* 70.1 (1983), pp. 41–55 (cit. on p. 42).

[Rub74] Donald B Rubin. "Estimating causal effects of treatments in randomized and nonrandomized studies." In: *Journal of educational Psychology* 66.5 (1974), p. 688 (cit. on pp. 40, 41).

[Rub05] Donald B Rubin. "Causal inference using potential outcomes: Design, modeling, decisions". In: *Journal of the American Statistical Association* 100.469 (2005), pp. 322–331 (cit. on p. 40).

[Sch78] Gideon Schwarz. "Estimating the dimension of a model". In: *The annals of statistics* (1978), pp. 461–464 (cit. on pp. 70, 71).

[Sha53] Lloyd S Shapley. "A value for n-person games". In: *Contribution to the Theory of Games* 2 (1953) (cit. on p. 127).

[SP08] Ilya Shpitser and Judea Pearl. "Complete identification methods for the causal hierarchy". In: *Journal of Machine Learning Research* 9 (2008), pp. 1941–1979 (cit. on pp. 35, 37–39).

[Sil+18] Tomi Silander, Janne Leppä-Aho, Elias Jääsaari, et al. "Quotient normalized maximum likelihood criterion for learning Bayesian network structures". In: *International conference on artificial intelligence and statistics*. PMLR. 2018, pp. 948–957 (cit. on pp. 70, 72).

[SM06] Tomi Silander and Petri Myllymaki. "A Simple Approach for Finding the Globally Optimal Bayesian Network Structure". In: *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence*. UAI'06. Cambridge, MA, USA: AUAI Press, 2006, pp. 445–452. ISBN: 0974903922 (cit. on p. 75).

[Sil+08] Tomi Silander, Teemu Roos, Petri Kontkanen, et al. "Factorized normalized maximum likelihood criterion for learning Bayesian network structures". In: *Proceedings of the 4th European workshop on probabilistic graphical models (PGM-08)*. 2008, pp. 257–272 (cit. on pp. 70–72).

[SRM10] Tomi Silander, Teemu Roos, and Petri Myllymäki. "Learning locally minimax optimal Bayesian networks". In: *International Journal of Approximate Reasoning* 51.5 (2010), pp. 544–557 (cit. on p. 72).

[SM05] Ajit P Singh and Andrew W Moore. *Finding optimal Bayesian networks by dynamic programming*. Carnegie Mellon University. Center for Automated Learning and Discovery, 2005 (cit. on p. 75).

[SV93] Moninder Singh and Marco Valtorta. "An algorithm for the construction of Bayesian network structures from data". In: *Uncertainty in artificial intelligence*. Elsevier. 1993, pp. 259–265 (cit. on pp. 51, 77).

[SGS90] P Spirtes, C Glymour, and R Scheines. "causality from probability, proceedings of Advanced Computing for the Social Sciences". In: *Williamsburgh, VA* (1990) (cit. on pp. 50, 58, 59).

[Spi01] Peter Spirtes. "An anytime algorithm for causal inference". In: *International Workshop on Artificial Intelligence and Statistics*. PMLR. 2001, pp. 278–285 (cit. on pp. 49, 64).

[Spi10] Peter Spirtes. "Introduction to Causal Inference". In: *Journal of Machine Learning Research* 11.54 (2010), pp. 1643–1662. URL: http://jmlr.org/papers/v11/spirtes10a.html (cit. on p. 44).

[SG91] Peter Spirtes and Clark Glymour. "An algorithm for fast recovery of sparse causal graphs". In: *Social science computer review* 9.1 (1991), pp. 62–72 (cit. on pp. 50, 58, 60).

[Spi+93] Peter Spirtes, Clark Glymour, Richard Scheines, et al. "Discovery algorithms for causally sufficient structures". In: *Causation, prediction, and search* (1993), pp. 103–162 (cit. on pp. 34, 50, 62, 107).

[SGS00] Peter Spirtes, Clark N Glymour, and Richard Scheines. *Causation, prediction, and search*. MIT press, 2000 (cit. on pp. 32, 34, 44, 45, 49, 58, 62).

[SR96] Peter Spirtes and Thomas Richardson. "A polynomial time algorithm for determining DAG equivalence in the presence of latent variables and selection bias". In: *Proceedings of the 6th International Workshop on Artificial Intelligence and Statistics*. Citeseer. 1996, pp. 489–500 (cit. on p. 48).

[SZ14] Peter Spirtes and Jiji Zhang. "A uniformly consistent estimator of causal effects under the k-triangle-faithfulness assumption". In: *Statistical Science* (2014), pp. 662–678 (cit. on p. 61).

[SZ16] Peter Spirtes and Kun Zhang. "Causal discovery and inference: concepts and recent methodological advances". In: *Applied informatics*. Vol. 3. 1. SpringerOpen. 2016, pp. 1–28 (cit. on pp. 44, 49).

[Suz93] Joe Suzuki. "A construction of Bayesian networks from databases based on an MDL principle". In: *Uncertainty in Artificial Intelligence*. Elsevier. 1993, pp. 266–273 (cit. on pp. 70, 71).

[Suz99] Joe Suzuki. "Learning Bayesian belief networks based on the minimum description length principle: basic properties". In: *IEICE transactions on fundamentals of electronics, communications and computer sciences* 82.10 (1999), pp. 2237–2245 (cit. on pp. 70, 71).

[TP02] Jin Tian and Judea Pearl. "A general identification condition for causal effects". In: *Aaai/iaai*. 2002, pp. 567–573 (cit. on p. 29).

[TP13] Jin Tian and Judea Pearl. "Causal discovery from changes". In: *arXiv preprint arXiv:1301.2312* (2013) (cit. on p. 36).

[TP15] Jin Tian and Judea Pearl. "On the Identification of Causal Effects". In: 2015. URL: https://api.semanticscholar.org/CorpusID:16596796 (cit. on p. 29).

[TT16] Sofia Triantafillou and Ioannis Tsamardinos. "Score-based vs Constraint-based Causal Learning in the Presence of Confounders." In: *Cfa@ uai*. 2016, pp. 59–67 (cit. on pp. 50, 74).

[TAS03] Ioannis Tsamardinos, Constantin F Aliferis, and Alexander Statnikov. "Time and sample efficient discovery of Markov blankets and direct causal relations". In: *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. 2003, pp. 673–678 (cit. on p. 62).

[TBA06] Ioannis Tsamardinos, Laura E Brown, and Constantin F Aliferis. "The max-min hill-climbing Bayesian network structure learning algorithm". In: *Machine learning* 65 (2006), pp. 31–78 (cit. on pp. 51, 77).

[Tsi+18] Konstantinos Tsirlis, Vincenzo Lagani, Sofia Triantafillou, et al. "On scoring maximal ancestral graphs with the max–min hill climbing algorithm". In: *International Journal of Approximate Reasoning* 102 (2018), pp. 74–85 (cit. on p. 78).

[VH15] Peter Van Beek and Hella-Franziska Hoffmann. "Machine learning of Bayesian networks using constraint programming". In: *Principles and Practice of Constraint Programming: 21st International Conference, CP 2015, Cork, Ireland, August 31–September 4, 2015, Proceedings 21*. Springer. 2015, pp. 429–445 (cit. on pp. 76, 105, 121, 133).

[VP91] T Verma and Judea Pearl. *Equivalence and synthesis of causal models Proceedings of the Sixth Annual Conference on Uncertainty in Artificial Intelligence*. 1991 (cit. on pp. 18, 23, 31, 33).

[Ver+17] Louis Verny, Nadir Sella, Séverine Affeldt, et al. "Learning causal networks with latent variables from multivariate information in genomic data". In: *PLoS computational biology* 13.10 (2017), e1005662 (cit. on pp. 50, 65, 98, 107).

[Vig15] Tyler Vigen. *Spurious correlations*. Hachette UK, 2015 (cit. on p. 26).

[YLW11] Jing Yang, Lian Li, and Aiguo Wang. "A partial correlation-based Bayesian network structure learning algorithm under linear SEM". In: *Knowledge-Based Systems* 24.7 (2011), pp. 963–976 (cit. on p. 78).

[YKU18] Karren Yang, Abigail Katcoff, and Caroline Uhler. "Characterizing and learning equivalence classes of causal DAGs under interventions". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 5541–5550 (cit. on p. 33).

[YM05] Sandeep Yaramakala and Dimitris Margaritis. "Speculative Markov blanket discovery for optimal feature selection". In: *Fifth IEEE International Conference on Data Mining (ICDM'05)*. IEEE. 2005, 4–pp (cit. on p. 62).

[ZL20] Benito van der Zander and Maciej Liśkiewicz. "Finding minimal d-separators in linear time and applications". In: *Uncertainty in Artificial Intelligence*. PMLR. 2020, pp. 637–647 (cit. on p. 97).

[ZOS22] Alessio Zanga, Elif Ozkirimli, and Fabio Stella. "A survey on causal discovery: theory and practice". In: *International Journal of Approximate Reasoning* 151 (2022), pp. 101–129 (cit. on pp. 35, 49).

[Zha08] Jiji Zhang. "Causal reasoning with ancestral graphs". In: *Journal of Machine Learning Research* 9 (2008), pp. 1437–1474 (cit. on p. 48).

[ZS08] Jiji Zhang and Peter Spirtes. "Detection of unfaithfulness and robust causal inference". In: *Minds and Machines* 18 (2008), pp. 239–271 (cit. on p. 61).

[ZS12] Jiji Zhang and Peter L Spirtes. "A transformational characterization of markov equivalence for directed acyclic graphs with latent variables". In: *arXiv preprint arXiv:1207.1419* (2012) (cit. on p. 48).

[ZZL05] Hui Zhao, Zhongguo Zheng, and Baijun Liu. "On the Markov equivalence of maximal ancestral graphs". In: *Science in China Series A: Mathematics* 48 (2005), pp. 548–562 (cit. on p. 48).

[Zhe+23] Yujia Zheng, Biwei Huang, Wei Chen, et al. "Causal-learn: Causal discovery in python". In: *arXiv preprint arXiv:2307.16405* (2023) (cit. on p. 107).