

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
SANTIAGO - CHILE



“LOG-LIKELIHOOD INVARIANCE IN MARKOV EQUIVALENT BAYESIAN NETWORKS”

SOFÍA ISIDORA RIQUELME FLORES

THESIS FOR THE DEGREE OF
CIVIL ENGINEER IN COMPUTER SCIENCE

Supervisor: Roberto Asín, Universidad Técnica Federico Santa María
Co-supervisor: Roberto Grossi, Università di Pisa
Co-advisor: Claudio Lobos, Universidad Técnica Federico Santa María

July - 2025



CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): ☒ Memoria o trabajo de título; ☐ Tesis de Postgrado;

Título del trabajo: "Log-Likelihood Invariance in Markov Equivalent Bayesian Networks"

Nombre del candidato(a): Sofía Isidora Riquelme Flores

Carrera / Grado: Ingeniería Civil Informática

Campus: Santiago San Joaquín ; **Departamento:** Informática

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Roberto Asín Achá, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL

☒ El trabajo **NO contiene información que amerite confidencialidad** y puede ser publicado de inmediato en repositorio con acceso abierto.

☐ El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (embargo) por:

☐ 6 meses; ☐ 12 meses; ☐ 2 años; ☐ 3 años; ☐ 5 años; ☐ 10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 30/07/2025

; Firma:

Estudiante o Candidato(a):

Fecha: 30/07/2025

; Firma:

Sofía

Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.

DEDICATION

To my parents.

ACKNOWLEDGMENTS

First and foremost, I would like to extend my sincerest gratitude to my supervisors Roberto Asín from Universidad Técnica Federico Santa María and Roberto Grossi from Università di Pisa. Roberto Asín has been an invaluable support for me in throughout this journey, and without his guidance I would certainly not be where I am now. I am specially indebted to Professor Roberto Grossi from Università di Pisa. Despite no formal obligation, he generously dedicated a substantial amount of time to this work and imparted an enormous amount of knowledge of this topic. His commitment to supporting my research is greatly appreciated. I am very grateful for the opportunity to work with him and Università di Pisa.

I'd also like to thank all of my friends who have been with me since the beginning, specially to Bayron Valenzuela and Gabriel Vergara. Thank you for all the laughter, sleepless nights, and always being a sympathetic ear. A special thanks also goes to my childhood friend Esperanza Escobar, with whom, despite the distance, we still remain as close as ever.

To my friends I've made during my time here in Italy. Specially to Jack and Luisa, thank you for being my rock, my source of laughter, and for sharing delicious food during this challenging time away from home. Though I've known them for a relatively short time, their friendship has become truly indispensable to me.

I would also like to express my deepest gratitude to my husband. Thank you for being there for me since the beginning of my degree, and for reminding me daily that everything will turn out okay in the end. Thank you for showing me another way to view life, and for bringing out the best in me. These years with you have truly been the best of my life.

My sincerest thanks to my sister Gabriela, for being my lifelong best friend. Her support and immaculate sense of humor have helped me through the toughest times.

Finally, I want to express my eternal gratitude to my parents. All my life, they've shown me the importance of hard work through their example, while providing unconditional support. These past months, while we've been apart, I'm especially grateful for all their love, the countless opportunities they've given me, and everything they have taught me. I want nothing more than to make them proud.

ABSTRACT

Abstract— Bayesian Networks are widely used to model probabilistic relationships between variables through DAGs. However, multiple Bayesian Networks can represent the same set of conditional independencies, forming what is known as a MEC. This presents a challenge in model evaluation, as networks within the same MEC are statistically indistinguishable based on observational data alone. This work investigates whether the log-likelihood function, a standard metric for assessing model fit, remains invariant across all Bayesian Networks in a given MEC. After learning the CPDAG of a Bayesian Network from data, a method is implemented to enumerate DAGs in the corresponding MEC, converting each DAG to a Bayesian Network and compute the log-likelihood for each structure using Bayesian parameter estimation. Structural modifications are also introduced to produce networks outside the original MEC, enabling comparison of log-likelihood values across different equivalence classes. Experiments conducted on benchmark networks (Asia, ALARM, Hepar2) confirm that log-likelihood values remain constant within each MEC and vary significantly when the structure is modified to break *v-structures*. These findings suggest that log-likelihood invariance can serve as a reliable indicator of MEC membership for Bayesian Networks, allowing for more efficient model evaluation and structure learning without full enumeration.

Keywords— Bayesian Networks, Markov Equivalence Class, Log-Likelihood

GLOSSARY

BN: Bayesian Network

CPDAG: Completed Partially Directed Acyclic Graph

CPT: Conditional Probability Table

DAG: Directed Acyclic Graph

MEC: Markov Equivalence Class

Contents

ABSTRACT	iv
GLOSSARY	v
FIGURE INDEX	vii
INTRODUCTION	1
CHAPTER 1: PROBLEM DEFINITION	3
1.1 Context	3
1.2 Current situation	3
1.3 Objectives	4
1.3.1 Main objective	4
1.3.2 Specific objectives	4
CHAPTER 2: CONCEPTUAL FRAMEWORK	5
2.1 Graphs	5
2.1.1 Preliminary Concepts	5
2.1.2 Directed Acyclic Graphs (DAGs)	6
2.2 Markov Equivalence Classes	7
2.3 Bayesian Networks	9
2.4 The Likelihood function	10
CHAPTER 3: SOLUTION PROPOSAL	12
3.1 Structure Learning and CPDAG estimation	14
3.2 DAG Enumeration	14
3.3 Parameter Learning and Likelihood Computation	15
3.4 Structural modifications	16
CHAPTER 4: SOLUTION VALIDATION	18
4.1 Validation Strategy	18
4.2 Datasets Used	18
4.3 Experiments	22
4.3.1 Asia Network	22
4.3.2 ALARM Network	24
4.3.3 Hepar2 Network	26
CHAPTER 5: CONCLUSION	28
BIBLIOGRAPHICAL REFERENCES	30

List of Figures

1	Problem tree. Source: Own elaboration	4
2	Example of a directed acyclic graph. Source: Own elaboration	5
3	Example of a <i>v-structure</i> in the DAG of Figure 2. Source: Own elaboration . . .	6
4	Example of an <i>essential graph</i> of the DAG in Figure 2. Source: Own elaboration	7
5	Example of UCCGs with the same number of edges but different MEC sizes. Source: [He et al., 2015]	9
6	Example of a Bayesian network. Source: [Koller and Friedman, 2010]	10
7	Two Markov equivalent Bayesian Networks with different edge orientations. Both belong to the same MEC, as they share the same skeleton and <i>v-structures</i> (highlighted in orange). Source: Own elaboration	13
8	Asia network. Source: [Scutari, 2010]	19
9	ALARM network. Source: [Scutari, 2010]	20
10	Hepar2 network. Source: [Scutari, 2010]	21
11	CPDAG obtained for the Asia network. Source: Own elaboration	22
12	Two DAGs belonging to the MEC represented by Figure 11. Both DAGs maintain the same <i>v-structure</i> , but the edge between ``bronc" and ``smoke" is reversed. Source: Own elaboration	22
13	Modified DAGs after breaking <i>v-structures</i> . DAGs 1 and 3 belong form a new MEC different from the original. Source: Own elaboration	23
14	Log-Likelihood values of all of the Asia DAGs. The yellow line remains invariant as all the DAGs belong to the original MEC, and the green line shows variations as the DAGs belong to a different MEC. Source: Own elaboration . .	24
15	CPDAG obtained for the ALARM network. Source: Own elaboration	25
16	Log-Likelihood values of all of the ALARM DAGs. The yellow line, representing DAGs from the same MEC, remains the same. The green line, representing the DAGs after structural modifications, show a significant variation in the log-likelihood value. Source: Own elaboration	25
17	CPDAG obtained for the Hepar2 network. Source: Own elaboration	26
18	Log-Likelihood values of all of the Hepar2 DAGs. The yellow line remains invariant for all DAGs of the same MEC, and the green line shows significant variation after structural modifications. Source: Own elaboration	27

INTRODUCTION

Understanding causal relationships between variables is a central goal across a variety of disciplines, from statistics and artificial intelligence to biology and social sciences. Probabilistic graphical models, particularly Bayesian Networks, are a powerful tool that help represent causal and probabilistic relationships, because they offer a structured, clear and complete model of the conditional dependencies and their conditional probabilities using Directed Acyclic Graphs (DAGs) [Koller and Friedman, 2010]. These networks enable reasoning over sets of random variables and are widely used in tasks such as inference, diagnosis, prediction, and decision-making under uncertainty [Pearl, 1988].

A central challenge that arises when using Bayesian Networks, is that different networks can represent the same information on conditional dependencies, forming what is known as a Markov Equivalence Class (MEC). Bayesian Networks within the same MEC are statistically indistinguishable using purely observational data. This presents a limitation in model evaluation and causal discovery, as many structurally distinct models are equally compatible with the data. As a result, there is a need for metrics that allow for comparison or evaluation across models within the same equivalence class. This raises a fundamental question: is it necessary to consider each individual network when evaluating models, or can the MEC be evaluated as a whole?

In addition, many structure learning algorithms, particularly those used in large scale applications, are designed to return a representation of the Markov equivalence class rather than a single Bayesian Network. This is because evaluating or searching over individual networks is often computationally infeasible, because the number of structures (represented as DAGs) in a MEC grows exponentially with the number of variables [Chickering, 2002]. As a result, having tools that operate effectively at the MEC level, such as log-likelihood invariance, can provide significant computational advantages and guide more informed structure selection within equivalence classes.

As mentioned above, the structure of a Bayesian Network is a DAG. Despite significant advancements in the field, enumerating all DAGs within a MEC remains computationally expensive. The number of possible structures grows exponentially with the number of undirected edges in the skeleton of the MEC, making full enumeration infeasible for large or complex models. Therefore, identifying a reliable metric that can characterize an entire MEC without exhaustive enumeration is a relevant and open problem.

One function widely used in statistical modeling and structure learning is the log-likelihood function, which quantifies how well a model explains a given dataset by measuring the proximity between the model's predicted distribution and the observed data [Friedman et al., 1997]. Because of its central role in evaluating model fit, the log-likelihood function is often used as a scoring function to evaluate the quality of different models. In this context, the log-likelihood function could offer an efficient alternative to compare or select

models without the need for full enumeration. If the log-likelihood value remains invariant across all Bayesian Networks in a MEC, it can act as a representative metric for the entire equivalence class, addressing the challenges outlined above.

Consider, for instance, a clinical decision support system using Bayesian Networks to predict patient outcomes based on diagnostic variables. If two models yield the same predictions and fit the data equally well, it becomes critical to know whether these models belong to the same Markov equivalence class, especially if one will be chosen for interpretation or further experimentation [Onisko et al., 1998]. A metric that remains invariant across all Bayesian Networks in a MEC, such as, the log-likelihood, can serve as a powerful proxy to identify when models are equivalent without needing to examine each network individually. This can prevent misinterpretations that arise from treating statistically indistinguishable models as meaningfully different, leading to more reliable conclusions in sensitive decision-making contexts.

This work is motivated by the need to understand whether the log-likelihood function remains invariant across all Bayesian Networks within a MEC, and how it behaves when small structural changes push a network outside its equivalence class. Specifically, the aim is to investigate whether the log-likelihood function can be reliably used to compare models without explicitly enumerating all Bayesian Networks in a MEC. If so, this could simplify model evaluation and open the door to new strategies for structure learning and model selection under equivalence.

CHAPTER 1

PROBLEM DEFINITION

1.1 Context

In the field of theoretical computer science, probabilistic graphical models are widely used to represent dependencies or causal relationships between variables. Among the existing models, Bayesian networks stand out due to their ability to represent both causal relationships and the probability distributions of these variables through Directed Acyclic Graphs (DAGs). These models are used in areas such as machine learning, bio-informatics, social sciences, medicine and more.

Due to the widespread use of these tools, there is a growing need for a method that allows us to evaluate how well a model represents an underlying structure in the data. This evaluation is crucial both for the theoretical development of these models and their practical applications.

1.2 Current situation

Due to the nature of probabilistic graphical models, more than one model can be used to represent the same dataset. That is, different structures can induce the same set of conditional independencies, forming what is known as a Markov Equivalence Class (MEC). When working with a MEC, the only existing way to assess the quality of all of the structures it contains (which are DAGs) is to enumerate all the DAGs within the class and evaluate them individually.

Existing methods for enumerating DAGs within a MEC are computationally expensive, because the number of DAGs grows exponentially with the number of nodes [Chickering, 2002]. For this reason, enumerating all the DAGs becomes infeasible beyond a certain number of variables. Hence, there is a need for a metric that can evaluate all models within the same equivalence class without requiring exhaustive enumeration.

This problem directly affects research in structure learning, especially in contexts where different classes need to be compared or learned models need to be validated. The lack of such tools limits the evaluation and comparison of structural models such as Bayesian Networks.

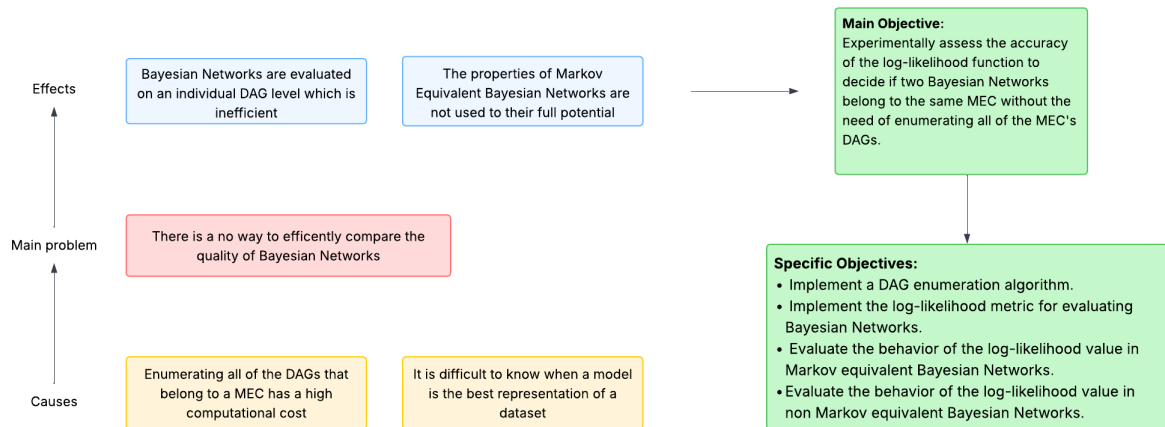


Figure 1: Problem tree. Source: Own elaboration

1.3 Objectives

1.3.1 Main objective

Experimentally assess the accuracy of the log-likelihood function to decide if two Bayesian Networks belong to the same MEC without the need of enumerating all of the MEC's DAGs.

1.3.2 Specific objectives

- Implement a DAG enumeration algorithm.
- Implement the log-likelihood metric for evaluating Bayesian Networks.
- Evaluate the behavior of the log-likelihood value in Markov equivalent Bayesian Networks.
- Evaluate the behavior of the log-likelihood value in non Markov equivalent Bayesian Networks.

CHAPTER 2

CONCEPTUAL FRAMEWORK

2.1 Graphs

2.1.1 Preliminary Concepts

Graphs are a discrete mathematics structure widely used in computer science. Formally, a graph G consists of a set of vertices and a set of edges, denoted as V and E , respectively.

In a graph, a *path* is a sequence of edges connecting different vertices, say $\{v_1, \dots, v_k\}$, such that one can go from v_1 to v_k if there exists an edge $v_i \rightarrow v_{i+1}$ or $v_i \leftarrow v_{i+1}$ for all $i = 1, \dots, k - 1$. A path is partially directed if at least one edge is directed. A path is directed (or undirected) if all edges are directed (or undirected). A cycle is a path that starts and ends at the same node. The *skeleton* of an arbitrary graph is an undirected graph that has the same vertices and edges as the original graph, ignoring directions.

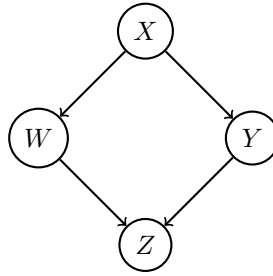


Figure 2: Example of a directed acyclic graph. Source: Own elaboration

A Directed Acyclic Graph (DAG) D is a directed graph that contains no directed cycles. Let V be the set of vertices of D and τ a subset of V . The *induced subgraph* D_τ of D on τ is defined as the graph where the vertices are from the set τ , and the edge set consists of all edges in D where both vertices are in τ . A *v-structure* is a three-vertex induced subgraph in D such that $v_1 \rightarrow v_2 \leftarrow v_3$, an example being the induced subgraph with vertices $\{W, Z, Y\}$.

A graph is called a *chain graph* if it contains no partially directed cycles. The isolated undirected subgraphs of a chain graph, after removing all directed edges, are called the *chain components* of the chain graph. A *chord* of a cycle is an edge connecting two non-consecutive vertices in a cycle. An undirected graph is called *chordal* if every cycle with four or more vertices has a chord.

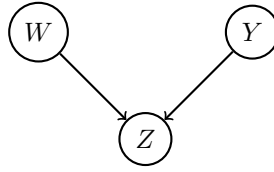


Figure 3: Example of a *v-structure* in the DAG of Figure 2. Source: Own elaboration

2.1.2 Directed Acyclic Graphs (DAGs)

In the context of Bayesian Networks, DAGs serve as the structural backbone that encodes the causal relationships between variables. DAGs are directed graphs with no cycles. In these graphs, nodes represent variables and directed edges indicate a dependency relationship between them. The absence of cycles ensures that it is not possible to return to a variable through a chain of dependencies, reflecting the principle that causality flows in a single direction.

Similarly, DAGs are widely studied in computer science due to their frequent appearance in various applications. Some examples include:

- **Task Scheduling with Order Constraints:** This problem involves managing a set of n tasks where some depend on the completion of others before they can begin. Hence, a strict order must be maintained to complete all tasks [Skiena, 1998]. A simple example can be seen in Excel spreadsheets: suppose we have four cells W , X , Y , and Z ; W and X contain numeric values, Y is the sum of $W + X$, and Z takes the value of Y . If the value of W changes, then Y and subsequently Z must be updated in that order.
- **Citation Networks of Scientific Articles:** These networks are inherently acyclic because articles cannot cite future work, thus preventing cycles [Clough et al., 2014]. These networks have been extensively studied as they form vast citation systems with rich interconnection histories.
- **Information Compression:** DAGs are also used in compression techniques, where sequence generation can be modeled using structures like tries. Some algorithms use DAGs to reduce storage space in these structures, such as finding the shortest path in a DAG [Carmona and Arroyuelo, 2022].

DAGs are powerful tools for encoding assumptions about data-generating processes. Beyond providing a clear visual representation of relationships between variables, DAGs also allow the identification of conditional independencies or dependencies within a system. This is especially useful in causal inference, where one can determine which variables should be conditioned on to identify direct causal effects and avoid bias.

While DAGs provide the graphical structure of a Bayesian network, it is the combination of structure and conditional probabilities that defines a complete Bayesian model. The remainder of this section addresses a key structural property of BNs: the fact that multiple DAGs can represent the same set of conditional independencies. This leads us to the concept of Markov Equivalence Classes (MECs), which play a central role in understanding the limits of structure-based model identification.

2.2 Markov Equivalence Classes

One of the key challenges when using Bayesian Networks in causal inference is that different structures (i.e., different DAGs) can represent the same conditional independence relationships among variables. In such cases, the DAGs are said to be *Markov equivalent*. That is, although the graphs have different structures, they imply the same conditional independencies and, therefore, the same data distribution.

The set of all DAGs that are Markov equivalent forms what is known as a *Markov Equivalence Class* (MEC). This phenomenon presents a significant limitation in causal inference because it is not possible to distinguish between different DAGs within the same class using observational data alone. To resolve this ambiguity, additional information is needed, such as temporal or experimental data that can help differentiate between the equivalent causal models [Spirtes, 2000].

In 1990, it was shown that two DAGs are Markov equivalent if and only if they share the same *skeleton* and the same *v-structures* [Verma and Pearl, 2022]. In 1997, it was proven that the MEC can be represented by a single *essential graph* [Andersson et al., 1997].

An *essential graph* of a DAG D , denoted C , is a graph that has the same skeleton as D , and includes a directed edge if and only if that direction appears in every DAG in the equivalence class of D [He et al., 2015].

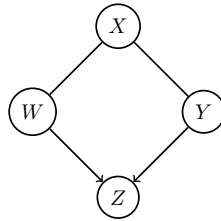


Figure 4: Example of an *essential graph* of the DAG in Figure 2. Source: Own elaboration

Since a MEC corresponds to the set of DAGs, a natural next question is how to determine the size of the MEC represented by its essential graph C . Clearly, if C is a DAG, the MEC has size 1; otherwise, C may contain more than one *chain component*, denoted $C_{\tau_1}, \dots, C_{\tau_k}$. In

[Andersson et al., 1997], it was shown that if C is the essential graph of D , then C is a *chain graph*, where each chain component C_τ of C is an undirected and *chordal* graph (UCCG), and τ is the set of vertices in the component. Any UCCG is an essential graph representing an MEC [Andersson et al., 1997]. Thus, the size of the MEC represented by C can be calculated by counting all DAGs in the MECs represented by the chain components using the following equation [He and Geng, 2008]:

$$sizeMEC(C) = \prod_{i=1}^k sizeMEC(C_{\tau_i})$$

Finally, to count the size of the MEC represented by a UCCG, all DAGs in the class can be generated. However, this can be very costly, as a large number of vertices in the UCCG may correspond to a huge and unmanageable number of DAGs.

In [He et al., 2015], two methods are presented for calculating the sizes of MECs with hundreds of vertices. The first method estimates the size using the number of vertices v and edges e in the UCCG [Gillispie and Perlman, 2002], but it only works in the following five cases:

- If $v = e - 1$, then $sizeMEC = e$
- If $v = e$, then $sizeMEC = 2e$
- If $v = \frac{e(e-1)}{2} - 2$, then $sizeMEC = (e^2 - e - 4)(e - 3)!$
- If $v = \frac{e(e-1)}{2} - 1$, then $sizeMEC = 2(e - 1)! - (e - 2)!$
- If $v = \frac{e(e-1)}{2}$, then $sizeMEC = e!$

In any other case, the size cannot be determined solely by the number of vertices and edges; it will depend on the essential graph structure.

For example, Figure 5 shows four UCCGs, the first two with five nodes and six edges, and the last two with five nodes and seven edges. None match any of the previous five cases. The corresponding MEC sizes are $sizeMEC(\mathcal{U}_{5,6}) = 13$, $sizeMEC(\mathcal{U}'_{5,6}) = 12$, $sizeMEC(\mathcal{U}_{5,7}) = 14$, and $sizeMEC(\mathcal{U}'_{5,7}) = 30$, illustrating that MEC sizes cannot be determined using only node and edge counts.

The second method involves partitioning the MEC into smaller subclasses. Using this and the previous method, a recursive algorithm was defined to calculate the size of an MEC that does not fit the five known cases.

From experimental results [He et al., 2015], the contributions can be summarized as follows:

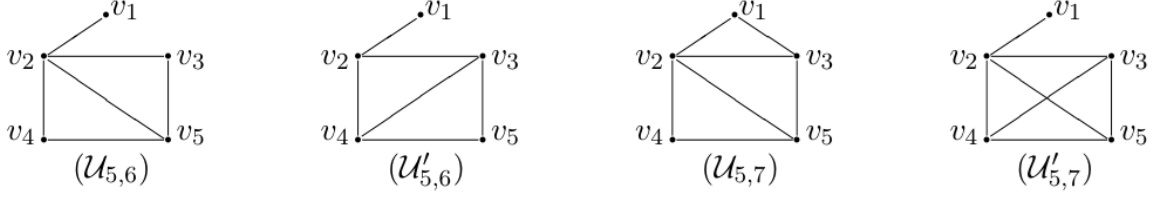


Figure 5: Example of UCCGs with the same number of edges but different MEC sizes. Source: [He et al., 2015]

1. The proposed approach can compute MEC sizes for UCCGs with fewer than 15 vertices in seconds.
2. For essential graphs with a sparsity constraint, the corresponding MEC sizes are nearly exponential in p .
3. In the full set of MECs of DAGs with p vertices, most graphs are semi-complete (around $p^2/4$ edges), and the MECs represented by these graphs have small average sizes. This explains why the average size of all MECs is small (around 3.7, as reported in [Gillispie and Perlman, 2002]) despite some sparse MECs being huge.

2.3 Bayesian Networks

Bayesian networks are probabilistic graphical models used to represent sets of variables and their conditional dependencies through a DAG. Formally, variables in a Bayesian network may represent observable quantities, unknown parameters, or hypotheses, and each edge represents a direct conditional dependency. Beyond representing dependency relationships, Bayesian networks also include the conditional probability distribution associated with each node.

For example, the network in Figure 6 represents the relationship between a student's intelligence, the difficulty of a course, the student's grade, their SAT performance, and the likelihood of receiving a recommendation letter from the professor. Each node in the network is associated with a conditional probability table that represents the probability of the event given its parent nodes. For instance, in the *Grade* node, the probability of obtaining grade A (denoted g^1) given that the student is not intelligent (i^0) and the course is not difficult (d^0) is 0.3. This feature makes Bayesian networks a powerful tool; in addition to representing causal and independence relationships, they also encode conditional probabilities.

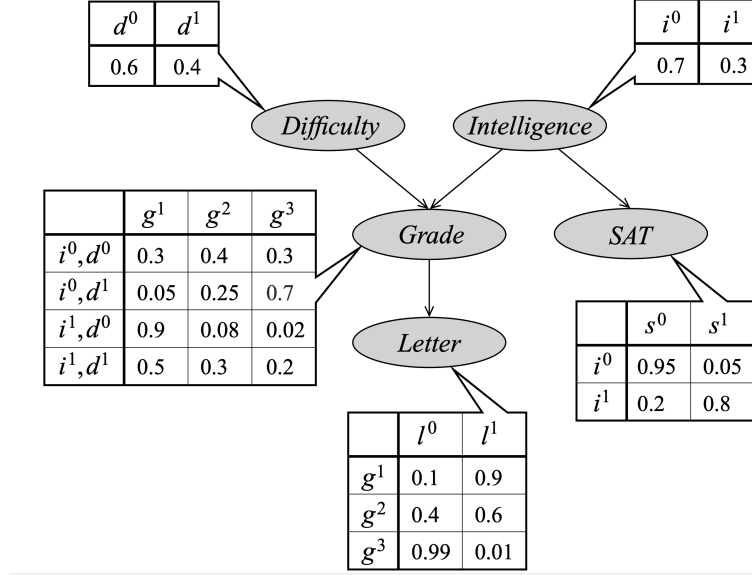


Figure 6: Example of a Bayesian network. Source: [Koller and Friedman, 2010]

2.4 The Likelihood function

The likelihood function measures how well a statistical model explains observed data. It does so by calculating the probability of observing the data under different parameter values of the model. Formally, let $\mathcal{D} = \{x^{(1)}, x^{(2)}, \dots, x^{(N)}\}$ be a dataset of N independent observations, where each $x^{(i)} = \{x_1^{(i)}, x_2^{(i)}, \dots, x_n^{(i)}\}$ represents a full assignment to all n random variables. Given a model with structure $\mathcal{G}$ parameterized by θ (e.g., conditional probability tables in a Bayesian Network), the likelihood function is defined as:

$$L(\theta; \mathcal{D}) = P(\mathcal{D} | \theta) = \prod_{i=1}^N P(x^{(i)} | \mathcal{G}, \theta)$$

However, working directly with the likelihood function can be computationally challenging, especially when the dataset is large, as it involves the product of many probabilities. To simplify computations, the log-likelihood function is often used. The log-likelihood function is the natural logarithm of the likelihood function:

$$\ell(\theta; \mathcal{D}) = \log L(\theta; \mathcal{D}) = \sum_{i=1}^N \log P(x^{(i)} | \mathcal{G}, \theta)$$

The log-likelihood function retains the same maxima as the likelihood function but transforms the product into a sum, making it more numerically stable and easier to differentiate. This property makes it particularly useful in optimization problems, such as parameter estimation using methods like maximum likelihood estimation (MLE) [Wakefield, 2013].

As stated before, the score of the log-likelihood function can be used to measure how well the model describes the data. However, direct interpretation of this score is difficult, but it can be used to compare models. A higher score means a better fit [Koller and Friedman, 2010].

CHAPTER 3

SOLUTION PROPOSAL

This work focuses on analyzing the behavior of the log-likelihood function, defined previously in Section 2.4. As shown in [Koller and Friedman, 2010], a Bayesian Network allows for the factorization of the joint distribution over variables, where the structure is given by a DAG and the local distributions are specified by Conditional Probability Tables (CPTs). Specifically, a distribution P factorizes according to the graph $\mathcal{G}$ the following way:

$$P(x^{(i)} \mid \mathcal{G}) = \prod_{j=1}^n P\left(x_j^{(i)} \mid \text{Pa}_{\mathcal{G}}(X_j)\right)$$

where $\text{Pa}_{\mathcal{G}}(X_j)$ denotes the set of parent variables of node X_j in the graph $\mathcal{G}$.

This factorization is essential, as it directly impacts the computation of the log-likelihood function for a dataset $\mathcal{D}$ with N observations:

$$\log \mathcal{L}(\theta; \mathcal{D}) = \sum_{i=1}^N \sum_{j=1}^n \log P\left(x_j^{(i)} \mid \text{Pa}_{\mathcal{G}}(X_j)\right)$$

This expression shows that the contribution of each variable to the log-likelihood depends entirely on its value and the value of its parents. In other words, the structure of the DAG, specifically the parent sets of each node, directly determines how the data is explained by the model.

In Section 2.2 it was shown that for two or more DAGs to be Markov equivalent, they must share the same skeleton and the same set of *v-structures* (Figure 3). This implies that all DAGs within a given MEC have identical collider configurations, meaning that the most critical conditional dependencies (where two variables simultaneously influence a third, without influencing each other) are preserved. An example is shown in Figure 7, where two DAGs over the node set $\{A, B, C, D, E, F, G\}$ differ in the orientation of the edge between A and B , yet preserve the *v-structure* formed by $E \rightarrow G \leftarrow F$. Since the skeleton and all *v-structures* remain unchanged, the corresponding Bayesian Networks belong to the same Markov equivalence class and encode the same conditional independence assumptions.

This observation leads to the intuition that the log-likelihood function should behave consistently across all Bayesian Networks belonging to the same MEC. Since the parent sets of each node that are involved in *v-structures* remain fixed within the class, it is hypothesized that

All Bayesian Networks within a given MEC should yield identical or nearly identical log-likelihood values when fitted to the same dataset.

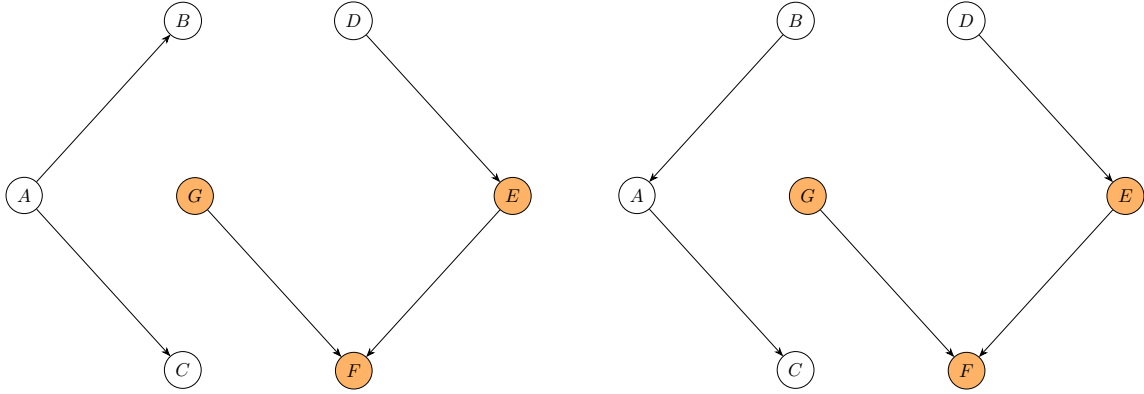


Figure 7: Two Markov equivalent Bayesian Networks with different edge orientations. Both belong to the same MEC, as they share the same skeleton and *v-structures* (highlighted in orange). Source: Own elaboration

In contrast, each structural change that affects a *v-structure* should lead to noticeably different values of the log-likelihood function.

This reasoning is what forms the foundation of the solution proposed in this thesis, that is, the validation of whether the log-likelihood values remain invariant across Markov equivalent Bayesian Networks, and to analyze how they change when structural modifications push a network outside the MEC. If confirmed, this property could be exploited for more efficient model evaluation, without requiring full enumeration of DAGs.

To empirically evaluate this hypothesis, the following methodology was applied:

- Simulate a dataset from a known DAG structure.
- Learn the CPDAG from the dataset using a structure learning algorithm.
- Enumerate all DAGs that are consistent with the CPDAG (therefore belong to the same MEC), or generate a representative subset when full enumeration is not computationally feasible.
- Convert each DAG into a Bayesian Network using parameter estimation.
- Introduce structural modifications to the CPDAG that intentionally alter its *v-structures*, thereby producing DAGs from a different MEC.
- Enumerate DAGs from this new MEC, and convert them to Bayesian Networks using parameter estimation
- Compute and compare the log-likelihood values to assess how they differ across MECs.

3.1 Structure Learning and CPDAG estimation

When learning Bayesian networks from a dataset, there are two key steps: structure learning and parameter estimation. These two key steps refer to the process of discovering the DAG that best represents the relationships between variables in the data. Parameter learning is usually performed afterward, and it consists of computing the probabilities that define the CPTs of each node based on the observed data.

In this work, the algorithm used for structure learning was the PC algorithm [Spirtes, 2000], specifically the implementation in the PGMPY library [Ankan and Textor, 2024]. This algorithm starts with a fully connected undirected graph, where all the variables are assumed to be potentially dependent. It then systematically tests for conditional independence between each pair of variables, conditioning on subsets of other variables. If two variables are found to be conditionally independent given some set, the edge between them is removed. The result is the undirected skeleton of the graph. Once the skeleton is determined, the algorithm identifies and orients *v-structures*. Additional rules are applied to orient as many edges as possible without introducing cycles or new *v-structures*, resulting in a CPDAG. This graph represents the MEC that encodes the same conditional independence relations.

The CPDAG obtained through this process serves as the basis for the DAG enumeration within the MEC.

3.2 DAG Enumeration

Since the structure of a Bayesian network is defined by a DAG, enumerating all DAGs in a given MEC allows us to construct and compare the corresponding Bayesian Networks and evaluate their log-likelihood values. There are a number of approaches to do this [Wienöbst et al., 2023, He et al., 2015, Ghassami et al., 2019], but a brute force approach proposed by [Chickering, 2002] was implemented for this step. The implementation can be seen in Algorithm 1.

This algorithm works by identifying all undirected edges of the CPDAG and generating all possible binary orientation combinations. For each combination it creates a new DAG by assigning directions accordingly, while making sure to preserve the *v-structures*. It then checks if the resulting graph is a valid DAG and whether it belongs to the MEC, and only the DAGs that satisfy both criteria are retained.

Input: A CPDAG, optional limit k on number of DAGs
Output: A list of DAGs consistent with the CPDAG (same MEC)
 Extract all undirected edges from the CPDAG
 Generate all binary orientation combinations for the undirected edges
 Initialize DAGs $\leftarrow$ empty list
foreach *orientation combination* **do**
 Initialize $G \leftarrow$ empty directed graph
 Add all nodes from the CPDAG to G
 Add all directed edges from the CPDAG to G
 foreach *undirected edge* (u, v) *with direction* d **do**
 if $d = 0$ **then**
 Add edge $u \rightarrow v$ to G
 end
 else
 Add edge $v \rightarrow u$ to G
 end
 end
 if G *is acyclic and preserves all v-structures of the CPDAG* **then**
 Add G to DAGs
 end
 if *limit* k *is defined and* $|DAGs| \geq k$ **then**
 break
 end
end
return DAGs

Algorithm 1: Enumerate DAGs consistent with a given CPDAG

While this brute-force enumeration strategy is conceptually straightforward, it is not computationally efficient. The number of valid DAGs grows exponentially with the number of undirected edges in the CPDAG, making it intractable for large or densely connected networks. Nevertheless, this method serves only as a means to support the experimental validation of the hypothesis of this work. The focus of this work lies not in proposing an efficient enumeration method, but in using this enumeration to analyze the behavior of the log-likelihood function across and between MECs.

3.3 Parameter Learning and Likelihood Computation

After enumerating the consistent DAGs, each DAG is converted into a Bayesian Network by estimating its parameters. This involves computing the Conditional Probability Tables (CPTs) that best explain the observed data. The method used is Bayesian Estimation, which, instead of relying purely on the data, incorporates prior beliefs and combines them with the observed frequencies to compute a posterior distribution over the parameters. This approach avoids assigning zero probability to unobserved configurations and yields more stable esti-

mates across multiple network structures, making it preferable for comparative analysis over Maximum Likelihood Estimation [Ji et al., 2015].

After parameter estimation was done for each consistent DAG so that they could become a Bayesian Network, the next step would be to compute the log-likelihood function for each of these. This was done again with the function already implemented by the PGMPY Python library [Ankan and Textor, 2024].

3.4 Structural modifications

Beyond comparing DAGs within the same MEC, this work also explores how structural changes, specifically, the alteration of *v-structures* affect the log-likelihood function. For this purpose, a method was implemented to deliberately break *v-structures*, therefore generating DAGs that belong to a different equivalence class. This can be seen in Algorithm

2

Input: DAG G , v-structure (C, P_1, P_2) , $\text{strategy} \in \{\text{"remove_parent1"}, \text{"remove_parent2"}, \text{"reverse_parent1"}, \text{"reverse_parent2"}\}$

Output: Modified Bayesian Network G'

Make a deep copy of G and call it G_{mod}

Let $C \leftarrow \text{child}$, $P_1 \leftarrow \text{first parent}$, $P_2 \leftarrow \text{second parent}$

if $\text{strategy} = \text{"remove_parent1"}$ **and** $P_1 \rightarrow C$ **exists in** G_{mod} **then**

 | Remove edge $P_1 \rightarrow C$ from G_{mod}

else if $\text{strategy} = \text{"remove_parent2"}$ **and** $P_2 \rightarrow C$ **exists in** G_{mod} **then**

 | Remove edge $P_2 \rightarrow C$ from G_{mod}

else if $\text{strategy} = \text{"reverse_parent1"}$ **and** $P_1 \rightarrow C$ **exists in** G_{mod} **then**

 | Remove edge $P_1 \rightarrow C$ from G_{mod}

if **no path from** C **to** P_1 **in** G_{mod} **then**

 | Add edge $C \rightarrow P_1$ to G_{mod}

end

else

 | Skip: adding $C \rightarrow P_1$ would create a cycle

end

else if $\text{strategy} = \text{"reverse_parent2"}$ **and** $P_2 \rightarrow C$ **exists in** G_{mod} **then**

 | Remove edge $P_2 \rightarrow C$ from G_{mod}

if **no path from** C **to** P_2 **in** G_{mod} **then**

 | Add edge $C \rightarrow P_2$ to G_{mod}

end

else

 | Skip: adding $C \rightarrow P_2$ would create a cycle

end

else

 | Print "Invalid strategy or edge does not exist."

end

Initialize empty Bayesian Network G'

Add all nodes from G to G'

Add all edges from G_{mod} to G'

return G'

Algorithm 2: break_v_structure: Modify a DAG to break a given v-structure

CHAPTER 4

SOLUTION VALIDATION

4.1 Validation Strategy

In order to validate the proposed hypothesis, a series of controlled experiments were designed to compare log-likelihood values under two conditions: Bayesian Networks whose structures belong to the same MEC, and Bayesian Networks whose structures belong to different MECs.

The strategy consisted of simulating datasets from a Bayesian network repository provided by the `bnlearn` library [Scutari, 2010] and applying structure learning techniques to recover their CPDAGs. From that CPDAG, their DAGs were then enumerated using Algorithm 1. Each DAG was converted into a Bayesian network by estimating its parameters (CPTs) using Bayesian Estimation and the log-likelihood was computed using the `PGMPY` library [Ankan and Textor, 2024]

To explore how structural changes affect the log-likelihood, specific *v-structures* were modified (reversed or removed), creating new DAGs from different MECs. These modified DAGs were also fitted to the same data and evaluated using the same procedure

4.2 Datasets Used

In this study, synthetic datasets were used. This decision was motivated by the need to have control over the data-generating process. It was crucial to control the number of samples generated in the dataset, because when using smaller sample sizes, the CPDAG learned from the data tends to have more uncertainty, leading to more undirected edges. This allows for more consistent DAGs to be possible instead of the alternative of having a single DAG when using large sample sizes. The use of synthetic datasets also allowed for more control over the structural changes induced.

While real-world datasets can add external validity, they introduce noise, unobserved confounding, and ambiguities in structure that would confound the specific objectives of this work. The use of synthetic data is therefore appropriate for isolating and analyzing the behavior of the log-likelihood function in a controlled, reproducible setting.

The experiments in this study use data generated from three well-known benchmark Bayesian Networks: Asia, ALARM, and Hepar2. These networks are not derived from raw observational datasets, but rather were manually constructed by domain experts to reflect realistic probabilistic dependencies in specific fields.

The Asia network [Lauritzen, 1988] is a compact Bayesian model consisting of eight binary variables related to respiratory diagnosis (e.g., tuberculosis, lung cancer, smoking, and x-ray results). Despite its simplicity, it captures key properties of probabilistic inference and is often used for testing structure learning algorithms due to its interpretable structure. It also allows for fast testing, given that it only has 8 variables.

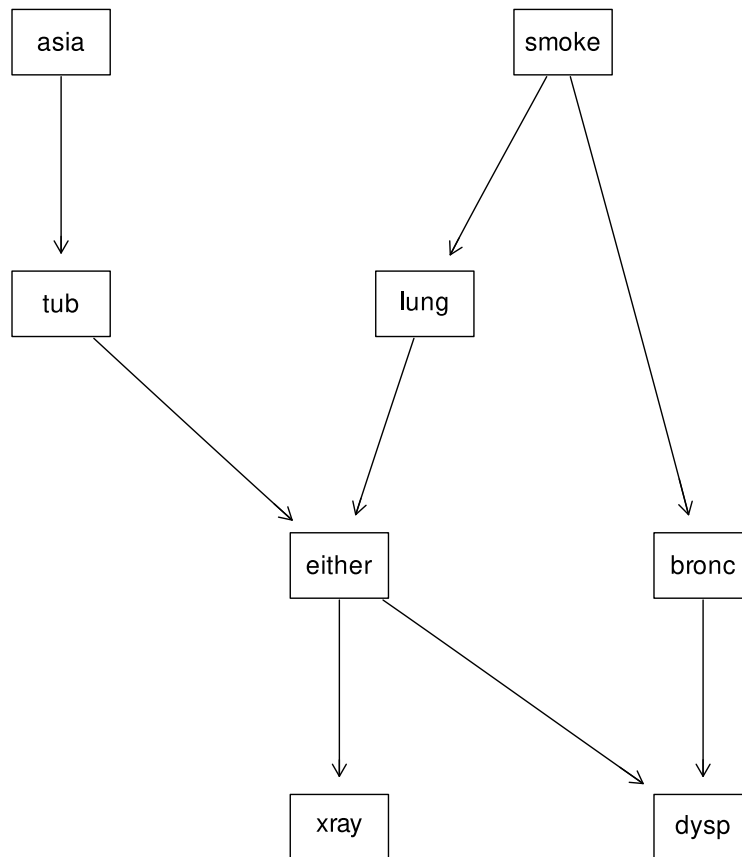


Figure 8: Asia network. Source: [Scutari, 2010]

The ALARM network [Beinlich et al., 1989] (A Logical Alarm Reduction Mechanism) was developed to model patient monitoring in an intensive care setting. It consists of 37 variables and 46 edges, and represents complex causal relationships among physiological states, diagnostic indicators, and monitoring instruments. ALARM is widely used in the literature to evaluate inference techniques and structural learning algorithms on more realistic, moderately sized networks.

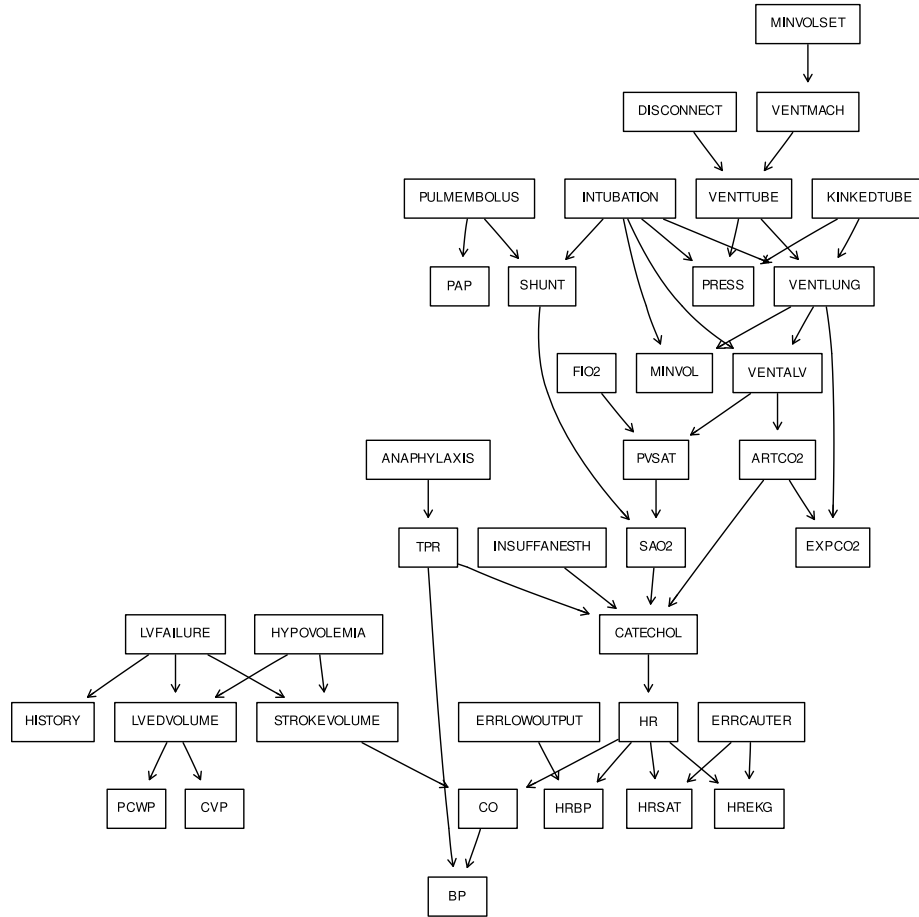


Figure 9: ALARM network. Source: [Scutari, 2010]

The Hepar2 network [Onisko et al., 1998] is a model originally designed for hepatitis. It consists of 70 variables and 123 edges. This network was manually constructed by domain experts to represent relationships between symptoms, test results, and underlying conditions. Due to its size and realistic medical domain, it is frequently used in the literature to test the scalability and accuracy of Bayesian Network methods [Jiang et al., 2022, Sambo and Franzin, 2018, Scutari, 2010].

4.3 Experiments

4.3.1 Asia Network

As mentioned in Section 4.2, the Asia network has 8 variables. After generating 1000 samples, the CPDAG obtained is shown in Figure 11. This CPDAG contains one bi-directed edge and two *v-structures*, which result in exactly two DAGs consistent with the equivalence class, shown in Figure 12.

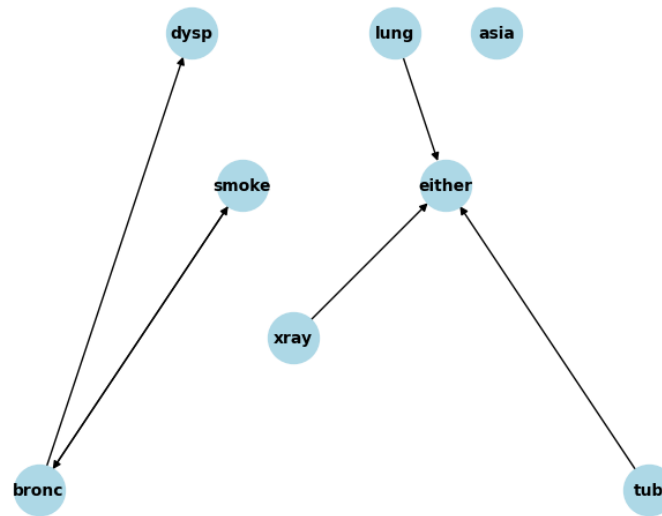


Figure 11: CPDAG obtained for the Asia network. Source: Own elaboration

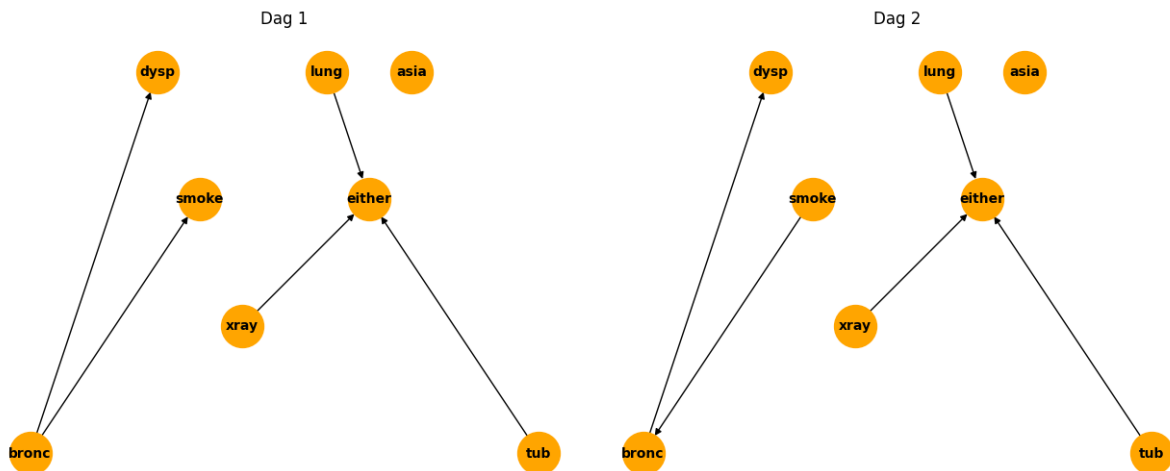


Figure 12: Two DAGs belonging to the MEC represented by Figure 11. Both DAGs maintain the same *v-structure*, but the edge between “bronc” and “smoke” is reversed. Source: Own elaboration

After applying Algorithm 2 to remove one parent in each *v-structure* of the original DAGs, four additional DAGs were created (Figure 13). These do not belong to the original MEC. Notably, Modified DAGs 1 and 3 form a new MEC, distinct from the original but shared between them.

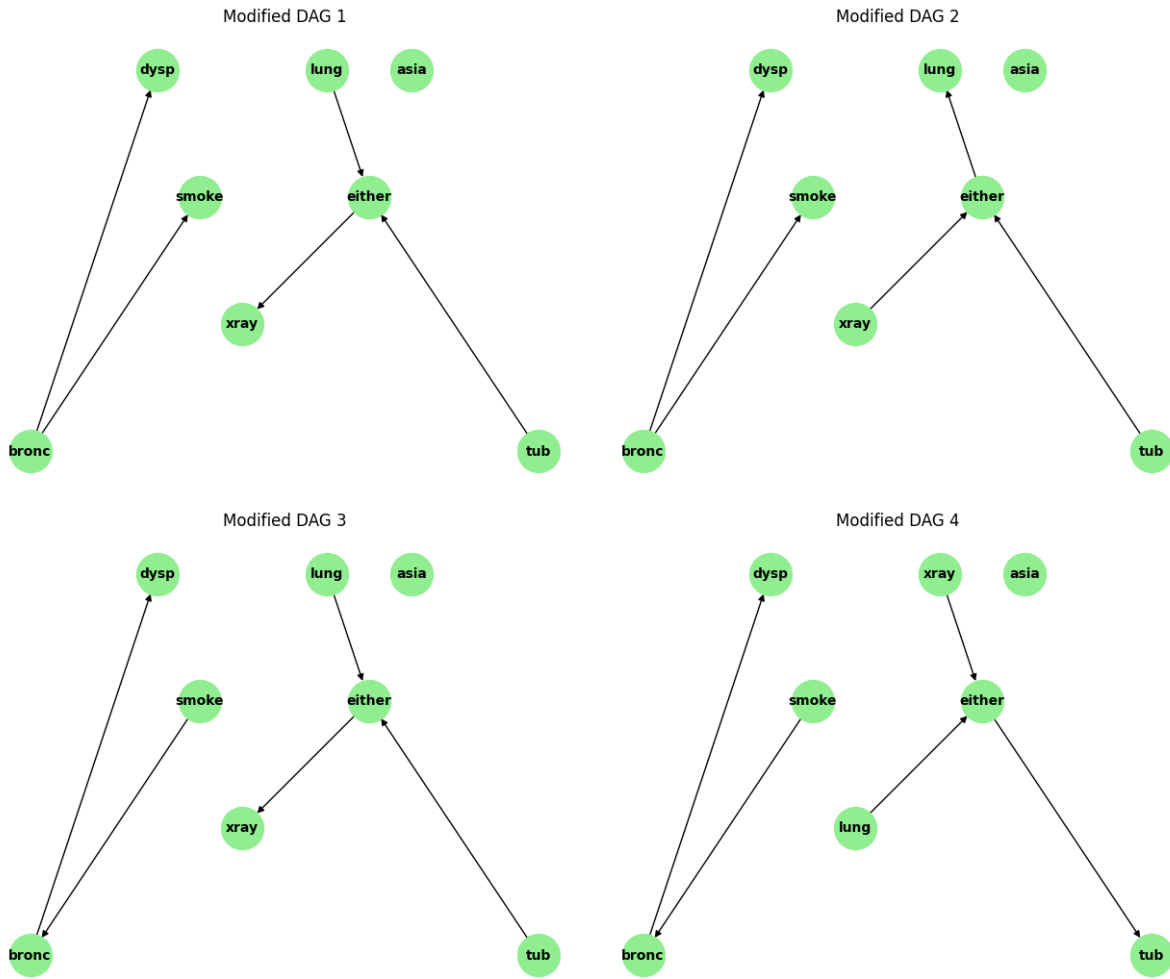


Figure 13: Modified DAGs after breaking *v-structures*. DAGs 1 and 3 belong form a new MEC different from the original. Source: Own elaboration

After applying parameter estimation, the log-likelihood values were computed for the six Bayesian networks derived from these DAGs. As shown in Figure 14, the two original Bayesian Networks, both from the same MEC, yield identical log-likelihoods, while Modified Bayesian Networks 1 and 3, which belong to a different but shared MEC, also yield identical values. This supports the hypothesis that log-likelihood is invariant within an MEC and sensitive to structural changes that move the graph out of it. While the Asia network is small, this initial result justifies further testing on larger networks.

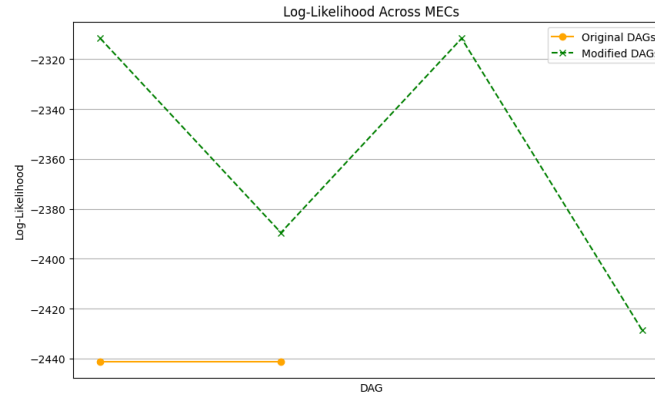


Figure 14: Log-Likelihood values of all of the Asia DAGs. The yellow line remains invariant as all the DAGs belong to the original MEC, and the green line shows variations as the DAGs belong to a different MEC. Source: Own elaboration

4.3.2 ALARM Network

The ALARM network contains 37 variables—a significant increase in complexity over the Asia network. A dataset of 1000 samples was generated, and the CPDAG learned using the PC algorithm is shown in Figure 15.

This CPDAG yielded 96 DAGs in its MEC. After altering a *v-structure* in each DAG (using both parent reversal strategies), 192 additional DAGs were generated from different MECs, for a total of 288 DAGs. Their log-likelihood values after converting each DAG to a BN are shown in Figure 16.

Interestingly, a subset of modified BNs formed a new MEC whose log-likelihood values were nearly identical to those of the original DAGs (differing by only 1.89). This small difference shows that two structurally distinct models (i.e., from different MECs) may still explain the data with almost equal fidelity. This finding illustrates a key challenge in model evaluation: structural changes may not always be easily distinguishable using likelihood-based methods alone.

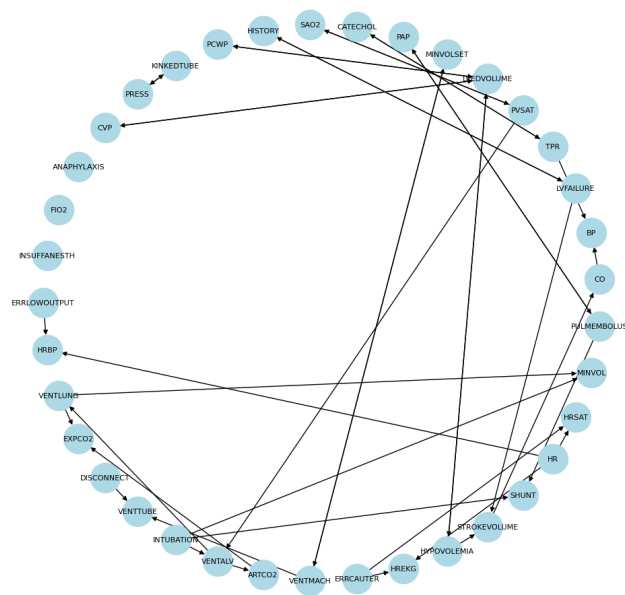


Figure 15: CPDAG obtained for the ALARM network. Source: Own elaboration

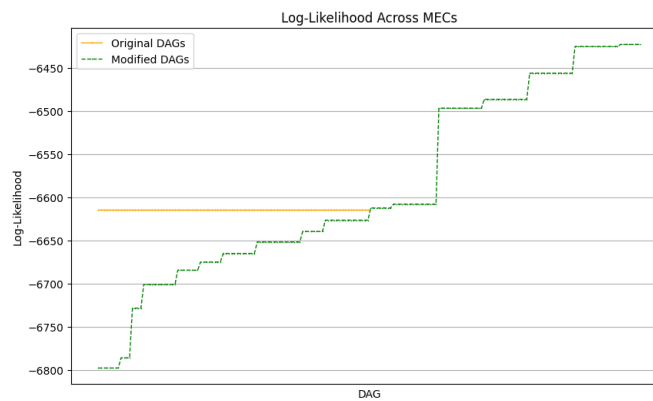


Figure 16: Log-Likelihood values of all of the ALARM DAGs. The yellow line, representing DAGs from the same MEC, remains the same. The green line, representing the DAGs after structural modifications, show a significant variation in the log-likelihood value. Source: Own elaboration

4.3.3 Hepar2 Network

The final network analyzed is Hepar2, which contains 70 variables, making it the largest network used in this study. A dataset of 500 samples was generated. The resulting CPDAG is presented in Figure 17.

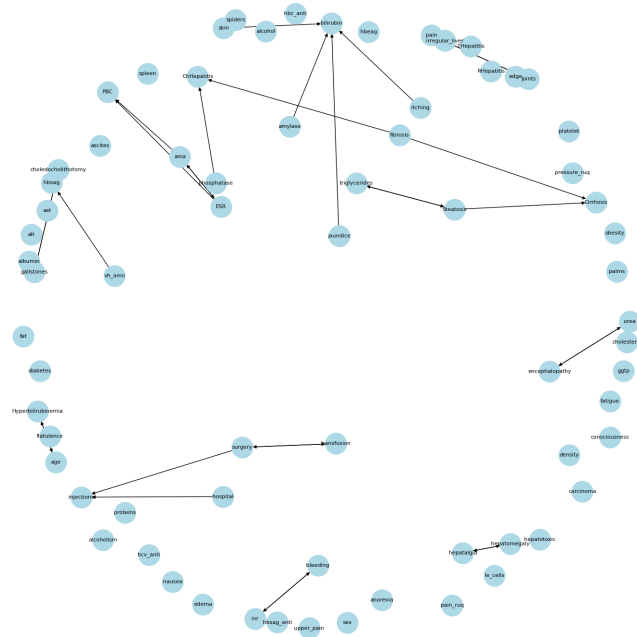


Figure 17: CPDAG obtained for the Hepar2 network. Source: Own elaboration

From this CPDAG, 512 DAGs were enumerated in the original MEC. After breaking each DAG's *v-structures*, 1024 additional DAGs were created from structurally distinct MECs. In total, 1536 DAGs were analyzed, and their log-likelihood values after converting them to BNs are shown in Figure 18.

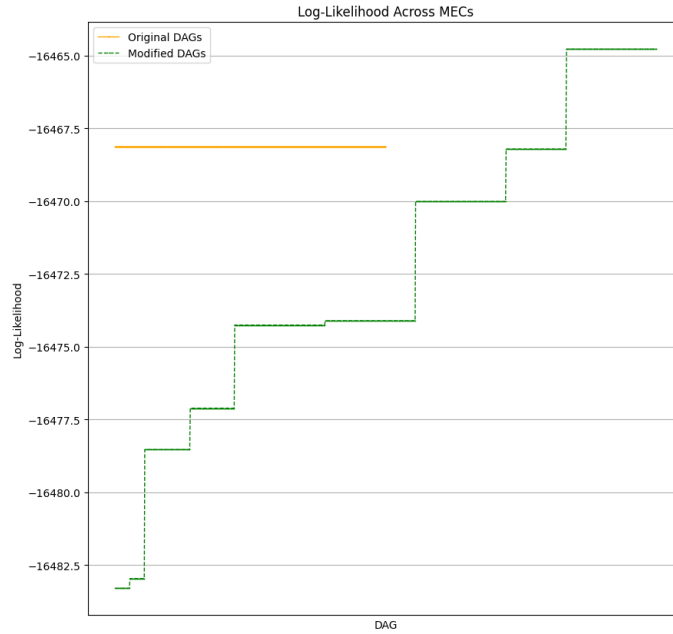


Figure 18: Log-Likelihood values of all of the Hepar2 DAGs. The yellow line remains invariant for all DAGs of the same MEC, and the green line shows significant variation after structural modifications. Source: Own elaboration

As expected, all Bayesian Networks within the original MEC yielded the same log-likelihood value. The modified Bayesian Networks, on the other hand, showed considerable variation. This reinforces the hypothesis: MEC membership is a strong predictor of log-likelihood behavior. Furthermore, this experiment demonstrates the scalability of the methodology to more complex networks with dozens of variables.

CHAPTER 5

CONCLUSION

This work examined whether the log-likelihood function remains invariant across all Bayesian Networks within a MEC, and how it behaves when structural modifications push a network outside of its equivalence class. Using synthetic datasets derived from the Asia, ALARM, and Hepar2 networks, the analysis focused on log-likelihood consistency across Bayesian Networks in the same MEC and on quantifying the changes resulting from deliberate alterations of *v-structures*.

The findings demonstrate that log-likelihood values remain identical for all Bayesian Networks within the same MEC, despite structural differences. This confirms that, from a statistical standpoint, such models encode the same conditional independence relations and therefore are equally supported by the data under a likelihood-based evaluation. In contrast, the networks that were minimally altered to fall outside the original equivalence class displayed measurable changes in log-likelihood. This divergence occurs because such structural modifications introduce new dependencies or eliminate existing ones, altering the joint probability distribution encoded by the network.

These results confirm that the log-likelihood function can serve as a reliable indicator of statistical equivalence. Bayesian Networks that belong to the same MEC not only represent the same independence assertions but also yield identical likelihood scores when fitted to the same data. Conversely, structural deviations that push a network outside the class tend to produce measurable differences in log-likelihood, thus offering a practical way to identify when models are no longer equivalent.

From a practical perspective, this has important implications for the evaluation of Bayesian Networks in real-world applications. In domains such as clinical decision making, automated diagnosis, and risk modeling, the ability to identify and work with MECs, rather than individual networks, can significantly reduce computational burden and improve the interpretation of the model. This approach supports more efficient structure learning by allowing practitioners to rely on representative structures without exhaustive enumeration, while also ensuring that different models are not mistakenly treated as distinct when they are statistically equivalent.

These insights open new opportunities for the development of scalable structure learning algorithms and highlight the value of log-likelihood as a diagnostic tool for navigating model equivalence in data-driven decision-making systems.

However, a key limitation of this study is that the analysis was conducted exclusively on fully specified Bayesian Networks, that is, a DAG with assigned parameters learned from data. This said, the findings about the log-likelihood invariance apply only when both the structure and the conditional probability distributions are available. This restricts the applicability of

the results in settings where only the DAG structure is available without the parameter estimates. In other words, this approach does not generalize directly to DAGs because it relies on evaluating full models. While structure alone determines conditional independencies, it is the combination of structure and parameters that shape the log-likelihood function. Therefore, while useful for Bayesian Networks, this method cannot currently be used as a structural test for MEC membership for DAGs.

Future work may include exploring other metrics beyond log-likelihood and integrating this evaluation approach into existing structure learning pipelines to enable more robust and scalable causal discovery. Another promising direction for future research involves exploring whether the observed invariance properties of the log-likelihood function can be extended to structural analysis alone while perhaps developing a method to computationally generating the data necessary for parameter estimation. This way the work could be extended to work with DAGs in general instead of only Bayesian Networks.

BIBLIOGRAPHICAL REFERENCES

- [Andersson et al., 1997] Andersson, S. A., Madigan, D., and Perlman, M. D. (1997). A characterization of markov equivalence classes for acyclic digraphs. *The Annals of Statistics*, 25(2):505–541.
- [Ankan and Textor, 2024] Ankan, A. and Textor, J. (2024). pgmpy: A python toolkit for bayesian networks. *Journal of Machine Learning Research*, 25(265):1–8.
- [Beinlich et al., 1989] Beinlich, I. A., Suermondt, H. J., Chavez, R. M., and Cooper, G. F. (1989). The alarm monitoring system: A case study with two probabilistic inference techniques for belief networks. In *AIME 89: Second European Conference on Artificial Intelligence in Medicine, London, August 29th–31st 1989. Proceedings*, pages 247–256. Springer.
- [Carmona and Arroyuelo, 2022] Carmona, G. and Arroyuelo, D. (2022). Implementacion de un algoritmo aproximado para la estructura partitioned elias-fano ϵ -optimal. *Tesis de Pregrado Acceso Abierto*.
- [Chickering, 2002] Chickering, D. M. (2002). Learning equivalence classes of bayesian-network structures. *Journal of machine learning research*, 2(Feb):445–498.
- [Clough et al., 2014] Clough, J. R., Gollings, J., Loach, T. V., and Evans, T. S. (2014). Transitive reduction of citation networks. *Journal of Complex Networks*, 3(2):189–203.
- [Friedman et al., 1997] Friedman, N., Geiger, D., and Goldszmidt, M. (1997). Bayesian network classifiers. In *Proceedings of the Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence*, pages 127–135.
- [Ghassami et al., 2019] Ghassami, A., Salehkaleybar, S., Kiyavash, N., and Zhang, K. (2019). Counting and sampling from markov equivalent dags using clique trees. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3664–3671.
- [Gillispie and Perlman, 2002] Gillispie, S. B. and Perlman, M. D. (2002). The size distribution for markov equivalence classes of acyclic digraph models. *Artificial Intelligence*, 141(1-2):137–155.
- [He et al., 2015] He, Y., Jia, J., and Yu, B. (2015). Counting and exploring sizes of markov equivalence classes of directed acyclic graphs. *The Journal of Machine Learning Research*, 16(1):2589–2609.
- [He and Geng, 2008] He, Y.-B. and Geng, Z. (2008). Active learning of causal networks with intervention experiments and optimal designs. *Journal of Machine Learning Research*, 9(Nov):2523–2547.
- [Ji et al., 2015] Ji, Z., Xia, Q., and Meng, G. (2015). A review of parameter learning methods in bayesian network. In *Advanced Intelligent Computing Theories and Applications: 11th*

- International Conference, ICIC 2015, Fuzhou, China, August 20-23, 2015. Proceedings, Part III 11*, pages 3–12. Springer.
- [Jiang et al., 2022] Jiang, J., Wen, Z., and Mian, A. (2022). Fast Parallel Bayesian Network Structure Learning. In *2022 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pages 617–627.
- [Koller and Friedman, 2010] Koller, D. and Friedman, N. (2010). *Probabilistic graphical models: principles and techniques*. Adaptive computation and machine learning. MIT Press, Cambridge, Mass., nachdr. edition.
- [Lauritzen, 1988] Lauritzen, S. L. (1988). Local computations with probabilities on graphical structures and their application to expert systems (with discussion). *J. Roy. Statist. Soc., B*, 50(2):251–263.
- [Onisko et al., 1998] Onisko, A., Druzdel, M. J., and Wasyluk, H. (1998). A probabilistic causal model for diagnosis of liver disorders. In *Proceedings of the Seventh International Symposium on Intelligent Information Systems (IIS—98)*, page 379.
- [Pearl, 1988] Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann.
- [Sambo and Franzin, 2018] Sambo, F. and Franzin, A. (2018). Bnstruct: an r package for bayesian network structure learning with missing data. *Last accessed May*.
- [Scutari, 2010] Scutari, M. (2010). Learning bayesian networks with the bnlearn r package. *Journal of Statistical Software*, 35(3):1–22.
- [Skiena, 1998] Skiena, S. S. (1998). *The algorithm design manual*, volume 2. Springer.
- [Spirtes, 2000] Spirtes, P. (2000). *Causation, Prediction, and Search*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, Mass, 2nd ed edition.
- [Verma and Pearl, 2022] Verma, T. and Pearl, J. (2022). Equivalence and synthesis of causal models. In *Probabilistic and Causal Inference: The Works of Judea Pearl*, pages 221–236. Association for Computing Machinery, New York, NY, USA.
- [Wakefield, 2013] Wakefield, Jon, o. (2013). *Bayesian and frequentist regression methods*, volume 23. Springer.
- [Wienöbst et al., 2023] Wienöbst, M., Luttermann, M., Bannach, M., and Liskiewicz, M. (2023). Efficient enumeration of markov equivalent dags. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 12313–12320.