universität wien

MASTERARBEIT | MASTER'S THESIS

Titel | Title

Distinguishing Cause and Effect by Analysis and Experimental Evaluation of the Heteroscedastic Noise Causal Method CLS-MML

verfasst von | submitted by

Suzana Marsela

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien | Vienna, 2025

Studienkennzahl lt. Studienblatt | Degree programme code as it appears on the student record sheet:	UA 066 921
Studienrichtung lt. Studienblatt | Degree programme as it appears on the student record sheet:	Masterstudium Informatik
Betreut von | Supervisor:	RNDr. CSc. Katerina Schindlerova Privatdoz.

Acknowledgements

I would like to express my deepest gratitude to my supervisor RNDr. Katerina Schindlerova (Hlaváčková-Schindler), CSc. Privatdoz., for her invaluable guidance, support, and encouragement throughout my thesis work. Her expertise, insightful feedback, and patience have been crucial in doing this project. I am especially thankful for the time she dedicated to enhancing ideas, and inspiring me to strive for excellence. This work would not have been possible without her mentorship.

A special thank you goes to my family, whose love and support have been very important to me. Their sacrifices, encouragement, and faith in my abilities have inspired me to pursue my goals. I would also like to thank my partner, Haris, for his unwavering love, patience, and constant support throughout this journey.

This thesis was completed as part of a prepared publication "Causal Location-Scale Noise Models by Minimum Message Length".

Abstract

We consider the challenge of causal inference in a bivariate setting using only observational data. A key component of this thesis is using the data values of two variables to infer which one is the cause and which is the effect. Inferring causal relationships, such as determining whether a particular lifestyle factor (like exercise) causes improved health or whether health influence lifestyle choices, is an ambitious but important task in many fields, like medicine, economics, agriculture and more. This thesis explores the utilization of bivariate causal methods based on an information-theoretic approach to decide causal direction between two variables.

We address the inference problem by using a Structural Causal Model (SCM) to represent the cause-and-effect relationship between variables, which allows the identification of the causal direction after estimating the model in both directions. Specifically, we employ a Location Scale Noise Model (LSNM) to model the effect variable where both the mean and variance of the noise term depend on the cause. To estimate this model we introduce a novel method where the cause variable follows Student's t-distribution. To infer the causality we utilize Kolmogorov Complexity (KC), particularly its approximation through Minimum Message Length (MML)principle. MML is used to find the model that minimizes the total length of the message needed to describe both the model and the data in both causal directions, choosing the one with the shorter length to be the cause. Therefore, we infer X causes Y in case the MML description is shorter to describe Y as a function of X than the inverse direction. The Causal Location Scale Minimum Message Length (CLS-MML) method, investigated in this work, integrates all these concepts to effectively distinguish between cause and effect among variables.

To illustrate the effectiveness of the CLS-MML method we tested against 13 benchmark datasets, including a real-world data called Tübingen causal pairs, which contain data from 99 distinct cause-effect pairs. Importantly, in this data basis, we have full access to the ground truth causal directions for all pairs, allowing us to validate our approach and motivate our choices. Our empirical evaluations demonstrated that our method achieves better precision results for some of the synthetic dataset and up to 70% accuracy on real dataset of Tübingen causal pairs. While exploring different optimization strategies to achieve maximum accuracy for our model, we observed that the Signal-to-Noise Ratio (SNR) of the dataset distribution played a crucial role. Additionally, the choice of hyper parameters for the Student's t-distribution, both those chosen initially and those optimized by the algorithm, significantly influenced the model's performance. Despite the simplicity of the model, our algorithm generally performs well on some synthetic datasets compared to Eight different methods and shows good results on Tübingen datasets, where it is competitive with state-of-the-art methods, achieving up to 70% accuracy.

Kurzfassung

Wir betrachten die Herausforderung der Kausalinferenz in einem bivariaten Szenario unter Verwendung rein beobachtender Daten. Ein zentraler Bestandteil dieser Arbeit ist die Nutzung der Datenwerte zweier Variablen, um zu bestimmen, welche von beiden die Ursache und welche die Wirkung ist. Kausale Zusammenhänge zu erschließen – wie etwa zu bestimmen, ob ein bestimmter Lebensstilfaktor (wie Sport) zu verbesserter Gesundheit führt oder ob Gesundheit die Lebensstilentscheidungen beeinflusst – ist eine ehrgeizige, aber wichtige Aufgabe in vielen Bereichen wie Medizin, Wirtschaft, Landwirtschaft und anderen. Diese Arbeit untersucht den Einsatz bivariater kausaler Methoden, die auf einem informations-theoretischen Ansatz basieren, um die kausale Richtung zwischen zwei Variablen zu bestimmen.

Wir adressieren das Inferenzproblem durch die Verwendung eines SCM, um die Ursache-Wirkungs-Beziehung zwischen Variablen darzustellen. Dadurch kann die kausale Richtung nach der Schätzung des Modells in beiden Richtungen identifiziert werden. Konkret verwenden wir ein LSNM, um die Wirkungsvariable zu modellieren, wobei sowohl der Mittelwert als auch die Varianz des Störterms von der Ursache abhängen. Um dieses Modell zu schätzen, führen wir eine neuartige Methode ein, bei der die Ursachvariable einer Student-t-Verteilung folgt. Zur Kausalitätsinferenz nutzen wir KC, insbesondere dessen Näherung durch das MML-Prinzip. MML wird verwendet, um das Modell zu finden, das die Gesamtlänge der Nachricht, die erforderlich ist, um sowohl das Modell als auch die Daten in beiden kausalen Richtungen zu beschreiben, minimiert. Die kürzere Beschreibung wird als Ursache identifiziert. Daraus schließen wir, dass X die Ursache von Y ist, falls die MML-Beschreibung kürzer ist, um Y als Funktion von X zu beschreiben, als in der umgekehrten Richtung. Die in dieser Arbeit untersuchte CLS-MML-Methode integriert all diese Konzepte, um effektiv zwischen Ursache und Wirkung von Variablen zu unterscheiden.

Um die Effektivität der CLS-MML-Methode zu veranschaulichen, haben wir sie an 13 Benchmark-Datensätzen getestet, einschließlich eines realen Datensatzes namens Tübingen Kausalpaar-Datensatz, der Daten von 99 verschiedenen Ursache-Wirkungspaaren enthält. Wichtig ist, dass in dieser Datenbasis die wahren kausalen Richtungen aller Paare vollständig bekannt sind, was uns ermöglicht, unseren Ansatz zu validieren und unsere Entscheidungen zu motivieren. Unsere empirischen Auswertungen zeigten, dass unsere Methode bei einigen synthetischen Datensätzen bessere Präzisionsergebnisse erzielt und eine Genauigkeit von bis zu 70% auf dem realen Tübingen-Datensatz erreicht.

Während wir verschiedene Optimierungsstrategien zur Maximierung der Genauigkeit unseres Modells untersuchten, beobachteten wir, dass das SNR der Datensatzverteilung eine entscheidende Rolle spielte. Darüber hinaus beeinflusste die Wahl der Hyperparameter der Student-t-Verteilung – sowohl der initial gewählten als auch der vom Algorithmus

optimierten – die Leistung des Modells erheblich. Trotz der Einfachheit des Modells schneidet unser Algorithmus im Vergleich zu 8 verschiedenen Methoden bei einigen synthetischen Datensätzen allgemein gut ab und zeigt gute Ergebnisse bei den Tübingen-Datensätzen. Hier ist er konkurrenzfähig mit den modernsten Methoden und erreicht eine Genauigkeit von bis zu 70%.

Contents

List of Tables

List of Figures

List of Algorithms

1. Introduction

In this chapter, we provide a clear introduction of this thesis and discuss its importance in the field of causality. Section 1.1 presents the problem statement, focusing on how to infer causality between two variables using our algorithmic framework. In Section 1.2, we outline our approach to addressing these challenges and detail our key contributions. Finally, Section 1.3 provides an outline of the steps taken in this work for the subsequent chapters.

1.1. Problem Statement

In this thesis, we will focus on inferring causality between two variables, a task commonly referred to as bivariate causality. While correlation can often be established through observational data, differentiating between cause and effect, remains a challenge in causal inference across various scientific domains. How and when can the causal structure be determined using only observational data? In many cases it is impossible to perform controlled experiments, and hence methods for discovering causal relations from pure data would be valuable. In our work, we will focus on bivariate causality in the field of statistical modeling and inference, where understanding the causal relationship between two variables is fundamental. The key idea of bivariate causality is to establish a direct connection of two variables, wherein the independent variable influences the dependent one. We investigate cases, whether a bivariate model is causally sufficient, i.e. we do not account for any outside factors, including confounding variables or biases, that could affect this direct relationship. The most common example is to decide whether altitude causes temperature, or vice versa, given only their joint measurements of both variables.

To tackle these causal relations we leverage the principles of information theory, particularly Kolmogorov Complexity [Kol68] and its practical approximation through Minimum Message Length principle [Wal05] . Initially Rissanen introduced MDL [Ris78] , but Wallace, C.S., Boulton, in their paper [WB68] introduced MML. MML is particularly valuable as it selects a model that is simpler but accurate. It can help us determine which of two potential causal directions (for example, whether variable X causes variable Y or vice versa) provides a better explanation of the data. We do this by comparing the complexity of the models and its parameters for each causal direction and selecting the direction that achieves a better fit with simpler explanations. By comparing this complexity corresponding to directions $X \to Y$ or $Y \to X$, MML helps select the direction that offers the most accurate explanation of the relationship between variables.

1.2. Objectives and Contributions

The aim of this master thesis is to infer the correct direction of causality between two variables using CLS-MML method across 13 datasets, having their ground truth, as part of the technical work of [HM24]. In the study of bivariate causality, understanding the role of noise in the observed data is essential. Traditional causal models often assume additive noise, where the relationship between cause and effect is perturbed by random variability. These models, typically are represented by Structural Causal Models SCM [Pea00]. However, we focus on studying Location-Scaled Noise Models LSNM that extend this framework by incorporating scaling functions into the noise term, allowing for more flexible causal relationships. Additionally, in many real-world applications, the observed data often deviate from a Gaussian distribution, exhibiting location-scale noise (often referred to as heavy-tailed noise) that violates standard regression assumptions. In such cases, robust regression methods, rather than traditional Gaussian noise models, are better suited for causal modeling.

Our work will focus on experimental analysis of the results obtained using the CLS-MML method [HM24], which employs robust regression with heavy-tail distribution of data, and we will compare its performance with other methods designed to address the same bivariate causality problem. This approach enables us to identify the most predictive causal direction between variables X and Y in the presence of scaled noise. To illustrate the performance of our method's we used 12 articially simulated datasets explained on the Section 4.1. Additionally, we used a real dataset Tübingen Cause-Effect Pairs, which have been gathered over the years to evaluate various bivariate causal discovery methods, see Section 4.2. We present the results of thorough tests to show that our method can compete with and even outperform some complex methods in finding causal relationships.

1.3. Outline

An overview of the main concepts and steps we followed in our thesis work is provided in Figure 1.1. In this workflow, the datasets are first preprocessed and then input into the CLS-MML algorithm from [HM24], where each variable pair is analyzed to infer the causal direction (either $X \to Y$ or $Y \to X$). During this process, statistical analysis is conducted to identify the best hyper parameters or determine the optimal Signal-to-Noise Ratio SNR threshold interval. Refer to Section 5.4 for details. The final output includes the causal direction for each pair, along with the best results achieved for each dataset based on the inferred causal relationships, see Section 7. Finally, we assess the performance of the algorithm using defined evaluation metrics such as Forced Decision Accuracy (Section 5.3.2) and Area Under the Decision Rate Curve (AUDRC) (Section 5.3.1). Section 8 discusses the results of the CLS-MML method, assessing its effectiveness, and Section 9 draws last conclusions.

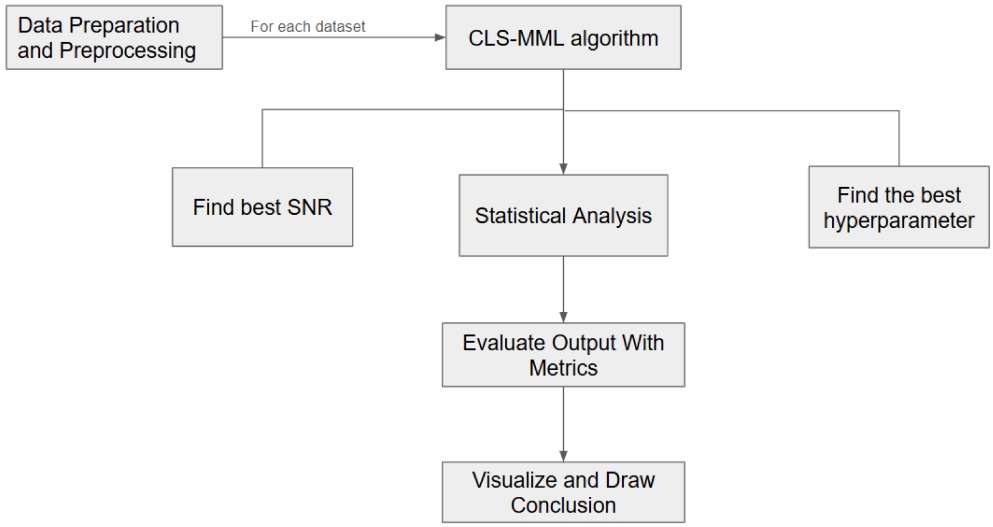

Figure 1.1.: Key steps of the thesis workflow.

2. Foundations

A preliminary background knowledge is essential to understand the results presented in this master thesis. This chapter will provide the foundational background for key topics, including bivariate causality, Structural Causal Models, and other models such as the Student's t-distribution regression model, Kolmogorov Complexity and its approximation through Minimum Message Length and more. Each concept will be linked by how it helps understand and explain cause-and-effect relationships in the data, giving a clear overall picture.

2.1. Bivariate Causality

In the field of statistical modeling and inference, exploring bivariate causality, the causal relationship between two variables is fundamental. In order to define a cause-effect relationship, it is required that these variables are statistically dependent and have a meaningful relationship, and that cannot be explained by correlation alone. For example, using only combined measures of temperature and altitude, one may examine if temperature is influenced by altitude or vice versa. This work will focus solely for the case where the causal models contain two variables. However, [Pea00] highlighted a serious challenge: without any prior information, distinguishing whether X causes Y (denoted as $X \rightarrow Y$) or Y causes X (denoted as $Y \rightarrow X$), is unattainable. One approach to address this challenge is to assume that the cause and effect are generated from a structural causal model [Pea00], which enables the model to be estimated in both directions before determining the causal direction.

2.1.1. Structural Causal Model

Structural Causal Models are a key framework used to represent the causal relation between variables from observational data. Each variable is associated with other variable using Structural Equation Modeling (SEM)) [Pea00]. These equations demonstrate the dependencies between variables and take the form of mathematical functions. In its most general form, an SCM, as described by [Pea00], typically consists of:

- **Variables**: Including observed variables (e.g., X, Y) and unobserved (latent) variables (e.g., ϵ, representing independent noise or error).
- **Functions**: Explanation of how each variable was created. For instance, an effect variable Y is modeled as a function f of a cause variable X and an independent noise term N (denoted as ϵ), such that $Y = f(X, N)$.

- **Causal Graph**: SCM are often represented using causal graphs, also known as Directed Acyclic Graphs (DAGs), where nodes represent variables and directed edges represent causal relationships between them. The graph structure reflects the causal dependencies among variables.

SCM can determine the true causal direction (X causes Y) under assumptions about the independence of causality and noise, where no backward model (Y causes X) holds under the same assumptions [PJS17]. However, [Moo+16] questions whether this asymmetry in the data-generating process—where Y is computed from X, but not the other way around—can be inferred solely from the joint distribution of X and Y. In contrast, [PJS17] argues that additional assumptions are required to detect causal structure, as such asymmetry cannot be determined from the joint distribution alone.

Building on this insight, recent literature has developed various approaches for identifying cause-effect relationships, which can be grouped into two branches: methods based in cause-effect asymmetry and methods that rely on the independence between the cause and the mechanism. This thesis adopts the latter approach, leveraging the independence assumption to develop a methods for identifying causal relationships. One key principle underlying this approach is expressed in **Postulate 1** about Structural Causal Models, known as the "Independence of Cause and Mechanism" [Sgo+15]. This principle assert that if $(X \rightarrow Y)$, the marginal distribution of the cause $P(X)$, and the conditional distribution of the effect given the cause, P(Y |X) are independent, i.e. P(X) contains no information about P(Y |X) and vice versa, since they correspond to independent mechanisms of nature. In simple terms, how the effect is generated from the cause does not give any information on how the cause itself is distributed. This independence does not hold in the opposite direction $(X \rightarrow Y)$, as P(Y) and P(X|Y) both inherit properties from P(Y |X) and P(X) and hence will contain information about each other. In this way, asymmetry between cause and effect has been introduced.

Postulate 1 is crucial to designing robust SCM and using them effectively for causal inference. An example that illustrates this principle can be found in nature as the relationship between sunlight and plant growth where, the former (the cause) influence the latter (the effect). The amount of sunlight a plant receives may vary due to other factor such as cloud coverage, the season change, and the mechanism of the plant growth itself involves other metabolic activities inside the plant. This means that sunlight exposure does not provide information about how it influences plant growth, and vice versa. They are considered separate mechanisms in nature. Considering the anti-causal direction the distribution of plant growth and the conditional distribution of sunlight exposure given plant growth contain information about each other. Knowing how plants grow may provide information into the amount of sunlight they receive, but it does not fully explain the effects of sunlight on their growth.

The principle of independent mechanisms is not just a theoretical postulate but it can also help design models in a way that improve their differentiability aspects. This connection will be further discussed later in this thesis.

2.1.2. Additive Noise Model

The Additive Noise Model (ANM), introduced by [Sgo+15], is a special case of a Structural Equation Model used for bivariate inference, where the influence of the noise on Y (the effect) is restricted to be additive. ANM is modeled as:

$$Y = f(X) + N \tag{2.1}$$

where Y is the sum of a functional relationship of the cause X plus some noise term N, with $X \perp\!\!\!\perp N$ (i.e., X and N are independent).

As discussed above, the assumptions derived for SCM apply similarly here. Specifically, for simple linear ANM, it has been demonstrated that in the case of non-Gaussian data, there is no backward model that can represent the cause modeled as a linear function of the effect with an additive independent noise [Sgo+15]. Rather, the noise in all of these backward models will depend on the effect. Therefore, we concentrate on the situation in which an asymmetry between the variables is introduced by the additivity constraint.

2.1.3. Location-Scale Noise Models

Location-Scale Noise Models, also known as Heteroscedastic Noise Models (HNMs), [Imm+22] generalize the concept of Additive Noise Models ANM by including both a location and a scale function into the noise component. In these models, the noise affecting the dependent variable is not only additive but also scaled by a function of the independent variable, allowing for more complex relationships between variables.

Location Scale Noise Model can be expressed as an SCM of the form:

$$Y = f(X) + g(X)N \tag{2.2}$$

where $g(X)$ adjusts the scale of the noise based on the cause X. LSNM are well-suited for modeling heavy-tailed noise because they allow for flexibility in variations of noise. This model is appropriate for modeling and analyzing relationships between pairs in our dataset choice, that have a heavy-tailed distribution, which is characterized by a high number of outliers than in normal distributions.

2.1.4. Student's t-distribution regression model

A Student's t-distribution regression model which we use a similar location-scale structure as in Eq. (2.2), however f, g are identity functions, the cause X follows a Student's t-distribution and N follows a Gaussian distribution. This approach is particularly effective for analyzing data with heavier tails, which a general normal distribution regression may not be able to address. Thus, the Student's t-distribution regression model can be seen as a generalization of the traditional linear regression model, where the location and scale parameters adapt to outliers or heavy-tailed error distributions.

Considering a SCM where ε follows a Student-t-distribution. Concretely we will express $f : X \to Y$ as a robust regression [HM24]:

$$Y = f(X) + \varepsilon = \beta_0 \mathbf{1}_n + x\beta + \varepsilon \tag{2.3}$$

with $y = (y_1, \ldots, y_n)$ and $x = (x_1, \ldots, x_n)$ and for each i

$$y_i \sim t(\beta_0 + x_i\beta, \tau, \nu). \tag{2.4}$$

$\mathbf{1}_n$ is the vector of ones of length n, is an error vector of length n, $\beta_0, \beta \in \mathbb{R}$. Under the Student-t regression model, the probability density of y_i

$$p(y_i|\mu_i, \tau, \nu) = \frac{\Gamma[(\nu+1)/2]}{\Gamma(\nu/2)\sqrt{\pi\nu\tau}}[1 + \frac{(y_i - \mu_i)^2}{\nu\tau}]^{-(\nu+1)/2} \tag{2.5}$$

The $\mu_i = \beta_0 + \beta x_i$, τ is a scale parameter and ν is a degree of freedom.

The parameter ν is crucial because it controls the heaviness of the tails in the distribution. As $\nu \to \infty$, the Student's t-distribution approaches the normal distribution which has lighter tails, while with smaller values of ν, the tails are heavier. By altering the distribution's shape, the Student's t-distribution can account for extreme outliers thanks to the degree of freedom, which is crucial in controlling variance.

We will use the notation LSNM for both Eq. (2.2) and model of Student's t-distribution regression in our thesis and when we discuss it, it will be clear from context which model concretely we mean.

2.2. Kolmogorov complexity

One of the important contributions to the theory of algorithmic complexity comes from Andrey Kolmogorov, a prominent figure in 20th-century mathematics. His work in this area laid the groundwork for what is known today as Kolmogorov complexity [Kol68].

Kolmogorov Theorem (1965) can be stated: There exists an optimal algorithm among those that decode strings from their descriptions (codes). For all strings, this algorithm allows codes as short as those allowed by other algorithms, up to an additive constant that is dependent on the algorithms but independent of the strings [Kol68]. The concept is tied to the idea of compressibility that a string with a simpler description is considered less complex.

The size of the smallest binary program, p^*, that, when executed on a universal Turing machine U, produces a finite binary string x before stopping is known as the Kolmogorov complexity of a string x, or $K(x)$ [Kol68]. Let le(.) be a function that maps a binary string $x \in \{0,1\}^n$ to its length, i.e.,

$$le : \{0,1\}^* \to N$$

.

Then

$$K(x) = le(p^*).$$

Mathematically, it is expressed as

$$K(x) = \min\{le(p) \mid p \in \{0,1\}^* \text{ and } U = x\}$$

Here, $\{0,1\}^*$ represents the set of all possible binary strings. The expression $U = x$ denotes that the universal Turing machine U generates outputs x when executing program p. The program (p^*) represents the most efficient algorithmic description of string x. The resulted program is of a lossless compression form of x and it contain no redundant information. Formally, $K(x|y)$ is the length of the shortest binary program (p^*) that generates x and halts when y is provided as an input to the program [Kol68]. This shows how the complexity of generating x changes when you already have some information y.

Additionally, Conditional Kolmogorov Complexity $K(x|y)$ is an extension of KC by taking into consideration some additional information such that it can measure the complexity of a string given some other string as input [LV93]. Hence, the amount of algorithmic information that y contains about x is defined as $I(y : x) = K(y) - K(y|x)$. Up to an additive constant term, than $I(X : Y) = I(Y : X)$.

Building on this idea, the concept of Kolmogorov complexity can also be applied to probability distributions. Specifically, the Kolmogorov Complexity of a probability distribution P, $K(P)$, is the length of the shortest program that outputs $P(x)$ to precision q on input (x, q) [LV93]. The conditional variant $K(P|Q)$ is defined similarly but with the additional information Q. Thus the Algorithmic Mutual Information (AMI) between distributions P and Q is $I(P : Q) = K(P) - K(P|Q)$, where Q^* is the shortest program for Q, refer to [JS10] for more details.

2.3. Causal Inference by Kolmogorov Complexity

This section outlines the approach to causal inference using Kolmogorov Complexity. We use it to investigate how to assess the independence between the distribution of a cause and the mechanism of its effect in causal inference.

[Sgo+15] articulated **Postulate 1 Independence of input and mechanism**:

If $X \to Y$, the marginal distribution of the cause, $P(X)$, and the conditional distribution of the effect given the cause, $P(Y \mid X)$, are independent. In other words, $P(X)$ contains no information about $P(Y \mid X)$ and vice versa, as they correspond to independent mechanisms of nature.

[JS10] articulated: **Postulate 2 Algorithmic Independence of Markov Kernels**:

If $X \to Y$, the marginal distribution of the cause $P(X)$ is algorithmically independent of the conditional distribution of the effect given the cause $P(Y|X)$, i.e., the algorithmic mutual information between the two will be zero, i.e.

$$I(P(X) : P(Y|X)) \stackrel{+}{=} 0$$

where I denotes algorithmic mutual information introduced in [JS10] in terms of Kolmogorov complexity.

The marginal distribution $P(X)$and the conditional distribution $P(Y|X)$ can be described independently to obtain the shortest description of the joint distribution $P(X, Y)$ [JS10]. The notation $\stackrel{+}{=} 0$ indicates that the relationship or equation is valid up to an additive constant. This postulate supports the causal inference framework by having a criterion for evaluating when causes and effects can be considered independent.

Building upon the concept of algorithmic independence, [LV93] suggests that the amount of algorithmic information contained in variable Y about variable X is expressed as:

$$I(Y : X) = K(Y) - K(Y|X^*) \tag{2.6}$$

where X^* is the shortest binary program for X. This quantifies the number of bits that can be saved in describing Y when the shortest description of X (represented by X^*) is already known. Algorithmic information is symmetric, meaning that $I(X : Y) = I(Y : X)$, where the $\stackrel{+}{=}$ denotes equality up to an additive constant. This symmetry leads to the concept of algorithmic mutual information. Called mutual information because it measures the amount of information shared between X and Y, considering both directions equally.

Based on both **Postulate 1** and **Postulate 2**, [Ste+10] proposed the following causal inference criterion:

If X is the cause of Y, then

$$K(P(X)) + K(P(Y \mid X)) \leq K(P(Y)) + K(P(X \mid Y)) \tag{2.7}$$

holds up to an additive constant.

This inequality implies that factorizing the joint distribution based on causality (i.e., using $P(X)$ and $P(Y|X)$) results in a simpler description compared to an alternative factorization that does not consider the causal relationship. By comparing the factorizations of their joint distribution, we can identify which variable influences the other, allowing us to infer the cause-effect relationship from observational data.

2.4. MDL and MML

While Kolmogorov Complexity remains incomputable due to the halting problem [Tur37], its application in causal inference provides a useful way to create a relationships between variables [LV93].

The halting problem is a well-known problem in computer science, stating that it is impossible to create a program that can determine whether for any given input it will stop executing or loop indefinitely. This problem impacts the Kolmogorov Complexity, as finding the shortest program to generate some output, requires to know if all the programs considered will eventually stop. Since KC is is not computable, there are practical approximations such as minimum message length and minimum description length that can be used in different contexts and explained below.

2.4.1. Minimum Description Length Principle

Inspired from KC principle, [Ris78] developed the minimum description length principle acknowledging that to encode data efficiently, one must account for restrictions or patterns within the data. In other words, data compression depends on the complexity of the model, as both need to be encoded. Given that Kolmogorov Complexity is not computable, MDL serves the purpose of providing an approximation for it. MDL does not directly compute the shortest program like KC does, but it offers a useful way for approaching the optimal solution. The algorithmic MDL, applies this principle using programs that generate the data and halt (lossless compressors) [Ris78].

Formally, given a set of models M and data D the best model $M \in \mathcal{M}$ is the one minimizing

$$L(D, M) = L(M) + L(D|M) \tag{2.8}$$

Where, L(M) represents the length in bits of the description of the model M and $L(D|M)$ represents the length in bits of the description of the data D when encoded (or compressed) using the model M. To sum up, MDL offers a practical way for encoding data with simplicity and effectiveness.

2.4.2. Minimum Message Length Principle

While many researchers rely on MDL, this thesis will focus on Minimum Message Length Principle [Wal05], which offers a more refined approach for our purposes. While MDL focuses on compressing data through the shortest possible model description, MML extends this concept by adding prior knowledge, aiming to minimize the total message length needed to encode both the data and the model. The minimum message length principle was introduced by C. S. Wallace and D. M. Boulton in the 1960s [Wal05], [WF87]. Since compression relies on underlying structure, the MML principle suggests that if a model compresses data effectively, it is likely capturing the true process that generates the data.

Here is how we can express the MML principle: The model that offers the most compact description of the data —both the model and the lossless encoding of the data— has a higher chance of being correct, even if the models have the same fit-accuracy to the observed data. In the MML principle, the message string of a observed data is called an explanation and has two components, the assertion and the detail. The assertion encodes a description of a statistical model (i.e., model structure and parameters) and the detail encodes the data using the model specified in the assertion. MML inference looks for the model that achieves the most data compression, meaning the model with the shortest two-part message. The fundamental idea of the MML approach is to balance the complexity of the model (measured by the code length of the assertion) with its ability to fit the observed data (measured by the code length of the detail).

Computing the exact MML code length for a given problem is generally NP-hard. However, there are numerous estimations that are available for calculating message lengths. One of the most widely used approximations is the MML87 formula [WF87],

which is based on a set of regularity conditions applied to the likelihood function and the prior distributions. A way is to use the small sample message length approximation, which remains robust even for highly peaked prior distributions [Wal05], making it suitable for Student-t priors as well.

3. Causal Inference by MML

As mentioned on the previous section, we know that KC is not computable. However, the criterion in equation 2.7 can be evaluated using the Minimum Message Length code length L, which serves as a computable approximation of Kolmogorov complexity. Instead of considering all possible programs, only those that generate X and eventually halt are taken into account, ensuring a lossless compression.

[HM24] defines $L(X)$ for the marginal distribution of X, analogously for $L(Y)$ and the $L(Y|X)$ as the length in bits of the description of the conditional distribution of Y given X. The criterion (2.7) changes to the form

$$L(X) + L(Y|X) \leq L(Y) + L(X|Y). \tag{3.1}$$

To compute the message length using MML87, one requires the negative log-likelihood function, a prior distribution over the parameters, and the corresponding Fisher information matrix [WMS18]. Wong et al. [WMS18] proposed an MML criterion for robust regression with Student-t distributed errors, which does not necessitates the calculation of the Fisher information matrix. From this criterion, we derive a criterion for $L(Y \mid X)$. On the other hand, $L(X)$ and $L(Y)$ are derived by adopting the approach from [MV17].

3.1. MML for $L(Y|X)$

In robust regression modeling, particularly with errors following a Student-t distribution, selecting an appropriate prior distribution is important for ensuring that the model is well-defined. To ensure a finite Minimum Message Length code for our robust regression models (2.7) with Student-t errors, it is necessary to properly bound the regression coefficients with a constrained parameter space using a hyper-ellipsoid approach, where the hyperparameter K regulates the bounding region's size, see [WMS18] for more details.

The value of K is computed as:

$$\hat{K} = \hat{\beta}_{ML}(\mathbf{X}^\top \mathbf{X})\hat{\beta}_{ML}, \tag{3.2}$$

where $\hat{\beta}_{ML}$ is the maximum likelihood estimate of β from 2.7 and X is the design matrix from the sample values of variable X.

Given a value of the hyper-parameter K, vector of parameters $\theta := (\beta_0, \beta; \tau, \nu)$, $\mu_i = \beta_0 + \beta x_i$, the total message length of Y and the parameters is [WMS18] :

$$I(\theta) = l(\beta_0, \beta, \tau) = \frac{1}{2}\log(1 + \frac{\hat{K}(\nu+1)}{3\tau(\nu+3)}) + \frac{n-1}{2}\log\tau + \sum_{i=1}^{n}\frac{w_i(y_i - \mu_i)^2}{2\tau}. \tag{3.3}$$

The derivations can be found in the [HM24].

We want to minimize (3.3) with respect to θ. [WMS18] propose to compute the maximum likelihood estimates in a Student-t regression model by the Expectation-Maximization (EM) algorithm. The EM algorithm is an iterative technique to find (local) maximum likelihood (or maximum) estimates of parameters in statistical models having unobserved latent variables. It can be used to find maximum likelihood estimates of parameters in models from an exponential family, see [Sun74].

The E-step of the EM algorithm when applied to minimize our MML criterion 3.3 computes :

$$\hat{w_i} = E(w_i|y_i, \theta) = \frac{\nu + 1}{\nu + \delta_i^2}. \tag{3.4}$$

The M-step of the EM algorithm computes

$$(\hat{\beta_0}, \hat{\beta}) = argmin_{\beta_0,\beta} \sum_{i=1}^{n} \hat{w_i}(y_i - \mu_i)^2 \tag{3.5}$$

$$\text{and} \qquad \hat{\tau} = argmin_{\tau} l(\tau, \hat{\beta_0}, \hat{\beta}). \tag{3.6}$$

The EM algorithm estimates $\hat{\beta}_0$, $\hat{\beta}$ and $l(\tau; \hat{\beta}_0, \hat{\beta})$ via numerical optimization. In summary, the MML estimates are computed by iterating the following steps until convergence [WMS18]:

1. Calculate the weights $\hat{w}_i$ using 3.4;
2. Update $\hat{\beta}_0$ and $\hat{\beta}$ based on the weights $\hat{w}_i$;
3. Update $\hat{\tau}$ using the current weights $\hat{w}_i$ and estimates $\hat{\beta}_0$ and $\hat{\beta}$.

In brief,[HM24] defines $L(Y|X)$ from (3.1) as $L(Y|X) := \min I(\beta_0, \beta, \tau)$ which is $I(\hat{\theta})$.

3.2. MML for $L(X)$ and $L(Y)$

To define the cost of codes of marginal distribution of $L(X)$ and $L(Y)$, [HM24] use the idea of [MV17]. Based on it, X and Y are normalized to be from the same domain. The data resolutions η_X and η_Y are determined for each set, where η_X represents the smallest difference between two data points in set X, and similarly, η_Y is defined for data set Y. While the distributions of X and Y are unknown, we know that both data sets contain n elements. Consequently, [MV17] encodes X as $L(X) = -n \log \eta_X$, and Y as $L(Y) = -n \log \eta_Y$.

3.2.1. Causal Inference by MML

[HM24] proposed a Causal Indicator that aims to infer the direction of causality between two variables X and Y based on the lengths of their descriptions and conditional distribution. The causality from X to Y is computed as:

$$\hat{\Delta}_{X\to Y} = \frac{L(X) + I(\hat{\theta})}{L(X) + L(Y)} \tag{3.7}$$

$L(X)$ and $L(Y)$ are the lengths in bits of the descriptions of the distributions of X and Y respectively. $I(\hat{\theta})$ is the length in bits of the description of the conditional distribution of Y given X.

- If $\hat{\Delta}_{X\to Y} < \hat{\Delta}_{Y\to X}$: It suggests that the causal direction is from X to Y. This means that the description of the joint distribution of X and Y is simpler when considering X as the cause and Y as the effect, compared to the reverse.
- If$\hat{\Delta}_{X\to Y} > \hat{\Delta}_{Y\to X}$: It suggests that the causal direction is from Y to X. This means that the description of the joint distribution of X and Y is simpler when considering Y as the cause and X as the effect, compared to the reverse.
- If $\hat{\Delta}_{X\to Y} = \hat{\Delta}_{Y\to X}$: It indicates that there is uncertainty in determining the direction of causality based on this indicator alone.

The measure $\hat{\Delta}_{X\to Y}$ is approximated from Kolmogorov Complexity , which is not explicitly defined in the given context but refers to a similar comparable measure of complexity in inferring causality from Y to X. Simply put, this causal indicator described by [HM24] offers a way to quantitatively find the direction of causality between two variables by looking at how simple their descriptions and conditional distributions are.

3.3. Signal noise ratio and MML

The signal-to-noise ratio of a random variable S to random noise N is given in literature, e.g. [Wik24b] as :

$$\mathrm{SNR} = \frac{\mathrm{E}[S^2]}{\mathrm{E}[N^2]}, \tag{3.8}$$

where E denotes the expected value, which in this case represents the mean square of N. Alternatively, the root mean squares can be used instead of the expected value:

$$SNR = \left(\frac{A_{\text{signal}}}{A_{\text{noise}}}\right)^2, \tag{3.9}$$

where A represents the Root Mean Square Error (RMSE).
Using $\hat{K}$ from (3.2) ,

$$\frac{\hat{K}}{\hat{\tau}} = \frac{\hat{\beta}_{ML}(\mathbf{X}^\top \mathbf{X})\hat{\beta}_{ML}}{\hat{\tau}} = nSNR, \tag{3.10}$$

where SNR is the empirical signal to noise ratio of the fitted Student-t regression model and $\hat{\tau}$ is the MML estimating of τ.

Thus,

$$SNR := n\frac{\hat{K}}{\hat{\tau}}. \tag{3.11}$$

This SNR can serve as a criterion for investigating for which amount of noise in a data can be the causal direction correctly identified.

Large SNR: For large signal strength the MML estimate, conditional on $\hat{\beta}$ and $\hat{\beta}_0$ is given by

$$\hat{\tau} = \frac{\sum_{i=1}^{n} w_i(y_i - \mu_i)^2}{n-2}. \tag{3.12}$$

Small SNR: For small signal strength the MML estimate, conditional on $\hat{\beta}$ and $\hat{\beta}_0$ is given by

$$\hat{\tau} = \frac{\sum_{i=1}^{n} w_i(y_i - \mu_i)^2}{n-1}. \tag{3.13}$$

The MML criterion proposed in [WMS18], which we apply to Eq. 2.1 as causal inference model, utilizes the estimate of τ for the small sample approximation (3.13). This approach is more robust and avoids resulting in negative code lengths for small SNR [WMS18].

4. Datasets

This chapter describes the datasets used in the experiments conducted in this thesis. Collecting real-world benchmark data can be challenging, primarily because ground truths are often unknown, and understanding the data-generating process to determine these truths is not straightforward. Therefore it is a common practice to evaluate the performance of methods using also simulated or synthetic data, where we have full control of the data-generating process and therefore know the ground truth. In this chapter we will include an explanation of 13 datasets from literature in detail. Among the essential data sets is the Tübingen dataset [Moo+16], the set of collected real data pairs with a known ground truth. These data sets will be used to assess our method.

4.1. Synthetic Datasets

The creation of the synthetic datasets can be done in many ways. The first sets of datasets considered are proposed by [TCV20] that consist of additive (AN, ANs), location scale (LS,LSs), and multiplicative (MNU) noise models with 100 pairs and 1000 data values. AN include nonlinear additive noise (AN) models of the form

$$Y = f(X) + E_Y \tag{4.1}$$

for some deterministic function f with $E_Y \sim N(0, \sigma^2)$, $X \sim N(0, \sqrt{2})$, and $\sigma \sim U\left[\frac{1}{5}, \sqrt{\frac{2}{5}}\right]$. In the additive noise model, f is an arbitrary nonlinear function created using Gaussian Processes (GP), as described by [RW06], with a Gaussian kernel with a one-bandwidth.

The location-scale generate data, where the cause determines both the effect's mean and variance, i.e.,

$$Y = f(X) + g(X)E_Y, \tag{4.2}$$

with E_Y and X following the same distribution as in the additive noise models. [RW06] propose another simulated dataset based on multiplicative models (MN) of the form

$$Y = f(X)E_Y, \tag{4.3}$$

where $f(X)$ is sampled as sigmoid functions and $E_Y \sim U(0, 1)$.

Since the functions in AN and LS may be non-injective, [RW06] include AN-s and LS-s to explore the behavior in injective cases. The datasets labeled with the suffix 's' and MN-U incorporate an invertible sigmoidal function, which makes the identification causality more difficult. These five datasets are well-suited for modeling with the location-scale noise model, which we assume they are an suitable test case for our methods. Additionally,

MNU serves as an interesting test case because it is not generated using Gaussian noise. All pairs are assigned equal weights, and the variable ordering is determined by a random coin flip, ensuring the datasets are balanced [RW06]. 4.1 illustrates 20 scatteplots which we generated from these simulated data pairs from [TCV20].

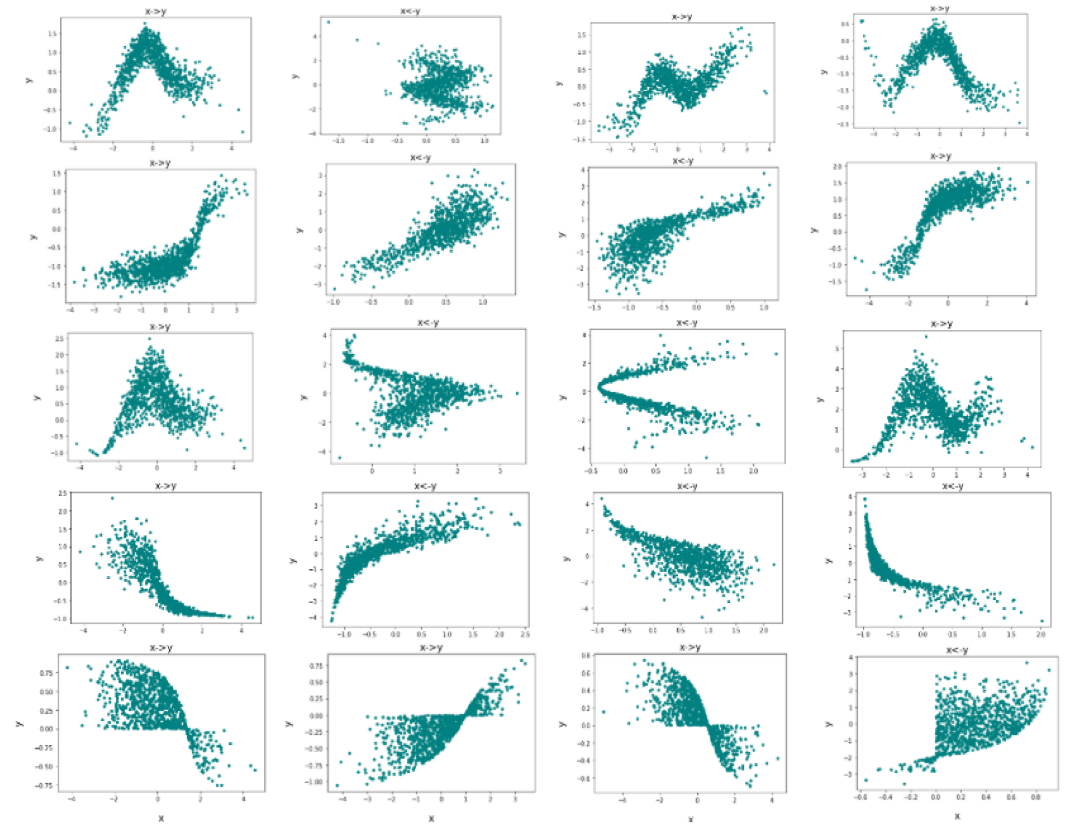

Figure 4.1.: Overview of the first four pairs from the AN/AN-s/LS/LS-s/MN-U cause-effect datasets from [TCV20]. Each figure displays a scatter plot where the x-axis represents the values of variable x and the y-axis represents the corresponding values of variable y. The title of each figure indicates the assumed causal relationship, specifying which variable is considered the cause and which is the effect.

Further, we consider another group of synthetic datasets SIM, SIM-c, SIM-ln, and SIM-G from [Moo+16], each having 100 pairs.

The datasets are simulated using the following relationships [Moo+16]:

- For cases **without a confounder** (SIM, SIM-ln, and SIM-G), the functions are:

$$X := f_X(N_X) \tag{4.4}$$

and

$$Y := f_Y(X, N_Y), \tag{4.5}$$

where N_X and N_Y represent noise terms sampled from random distributions, generated by using functions sampled from a GP to map a standard normal distribution.

- For the case **with a confounder** (SIM-c), the functions are:

$$Z := f_Z(N_Z), \tag{4.6}$$

$$X := f_X(N_X, N_Z), \tag{4.7}$$

and

$$Y := f_Y(X, N_Y, N_Z) \tag{4.8}$$

where the confounder Z also depends on a noise term N_Z, and the relationships between X, Y, and Z are modeled using randomly drawn functions from a GP.

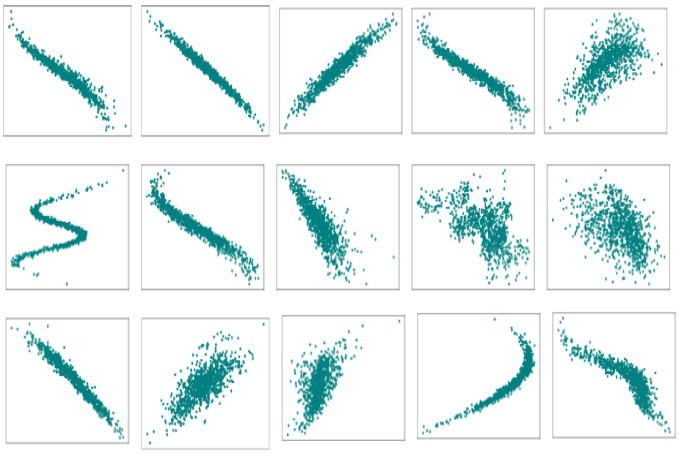

Figure 4.2.: Overview of the some cause-effect pairs in simulation scenario for SIM-G dataset. The x-axis represents the values of x and y-axis represents the values of variable y.

In the SIM-ln dataset, models are subjected to low levels of noise, while the SIM-G dataset utilizes approximately Gaussian non-linear additive noise generative models, which is expected to favor methods assuming Gaussianity [Moo+16]. We would expect our algorithm to not perform ideally in the case where a confounder is present, as it does not account for confounding variables. Refer to the figure 4.2 , which provides examples of Sim-G simulated data pairs. Additionally, for the low noise and Gaussianity scenarios

it will be important to observe whether our algorithm will perform better, or if other methods assuming Gaussianity will outperform it.

And finally, we decribe more complex synthetic datasets, namely Multi, Net from [Gou+18] and Cha from [GSB19]. The data set Multi contains 300 artificial pairs generated with linear and polynomial techniques. This dataset is generated using pre-additive noise models ($Y := f(X + N_Y)$), post-additive noise models , pre-multiplicative noise models ($Y := f(X \times N_Y)$), and post-multiplicative noise models ($Y := f(X) \times N_Y$). The pairs in the Net dataset contain 300 pairs which are generated using neural networks with random weights and random distribution for the cause X such as exponential, gamma, log-normal, or Laplace. The Cha benchmark includes 300 continuous variable pairs selected from the ChaLearn Cause-Effect Pairs Challenge, with pairs with label $+1$ ($X \to Y$) and -1 ($Y \to X$) [GSB19]. For all variable pairs, the sample size n is up to 1,500. These datasets do not have any associated weights for their data pairs.

4.2. Real-World Datasets

For a real-world benchmark, we select the Tübingen cause-effect pairs from [Moo+16] to compare the performance of the methods. This dataset consists of cause-effect pairs collected from a variety of sources across different domains. This dataset comprises in total 108 pairs of samples from two statistically dependent random variables, with one variable being identified as the cause of the other. We selected 99 out of the 108 pairs that consist of one-dimensional variables, excluding pairs 47, 52–55, 71, 105, and 107. These pairs are drawn from 37 distinct datasets across various domains: climate, medicine data and more. Each pair in this benchmark is associated with a weight that scales down the results of pairs with similar properties. For more details, refer to Table 4.1. As a result, the evaluation outcomes for this benchmark are also weighted based on these assigned weights. The task is to determine, based solely on the observed samples, which of the two variables in each pair is the cause and which one is the effect. The datasets were selected with expectation that the ground truth would be accepted as given. For instance, the first pair includes measurements of altitude and mean annual temperature from over 300 weather stations in Germany. It is clear that altitude influences temperature, rather than the reverse. Despite this, part of the statistical dependencies may also arise due to hidden common causes and selection bias [Imm+23].

4.2.1. One pair example

This section will focus on a specific subset of the Tübingen pairs dataset, presenting a detailed description of the cause-effect pairs and explain the determination of the causal relationships within each other. We made scatterplots of the pairs referred as pair0038 through pair0041. The horizontal axis denotes the cause and the vertical axis denotes the effect 4.3. For additional details on the other pairs, see [Moo+16].

Pair	Var 1	Var 2	Dataset
pair0001	Altitude	Temperature	DWD
pair0002	Altitude	Precipitation	DWD
pair0003	Longitude	Temperature	DWD
pair0004	Altitude	Sunshine hours	DWD
pair0005	Age	Length	Abalone
pair0006	Age	Shell weight	Abalone
pair0007	Age	Diameter	Abalone
pair0008	Age	Height	Abalone
pair0009	Age	Whole weight	Abalone
pair0010	Age	Shucked weight	Abalone
pair0011	Age	Viscera weight	Abalone
pair0012	Age	Wage per hour	Census income
pair0013	Displacement	Fuel consumption	Auto-mpg
pair0014	Horse power	Fuel consumption	Auto-mpg
pair0015	Weight	Fuel consumption	Auto-mpg
pair0016	Horsepower	Acceleration	Auto-mpg
pair0017	Age	Dividends from stocks	Census income
pair0018	Age	Concentration GAG	GAGurine (from R package MASS)
pair0019	Current duration	Next interval	Geyser
pair0020	Latitude	Temperature	DWD
pair0021	Longitude	Precipitation	DWD
pair0022	Age	Height	Arrhythmia
pair0023	Age	Weight	Arrhythmia
pair0024	Age	Heart rate	Arrhythmia
pair0025	Cement	Compressive strength	Concrete data
pair0026	Blast furnace slag	Compressive strength	Concrete data
pair0027	Fly ash	Compressive strength	Concrete data
pair0028	Water	Compressive strength	Concrete data
pair0029	Superplasticizer	Compressive strength	Concrete data
pair0030	Coarse aggregate	Compressive strength	Concrete data

Table 4.1.: Overview of some pairs from real dataset Tübingen, where you can see the variables from diffrent domains and the ground truth causal direction. For the complete table, see [Moo+16]

D.10 Pima Indians Diabetes

This dataset, accessible from the UCI Machine Learning Repository [BL13], was collected by the National Institute of Diabetes and Digestive and Kidney Diseases in the United States to predict the onset of diabetes mellitus in a high-risk population of Pima Indians living near Phoenix, Arizona. The cases in this dataset were chosen according to certain standards, including being female, at least 21, and of from Pima Indians descent. As a result, there can be an age-related selection bias. As described by [Moo+16], the dataset is explained as follows:

pair0038: Age –> Body Mass Index

Body Mass Index (BMI) is defined as the ratio of weight (in kg) to the square of height (in m). Clearly, age cannot be caused by BMI. However, since age influences both height and weight, it is reasonable to say that age causes BMI [Moo+16].

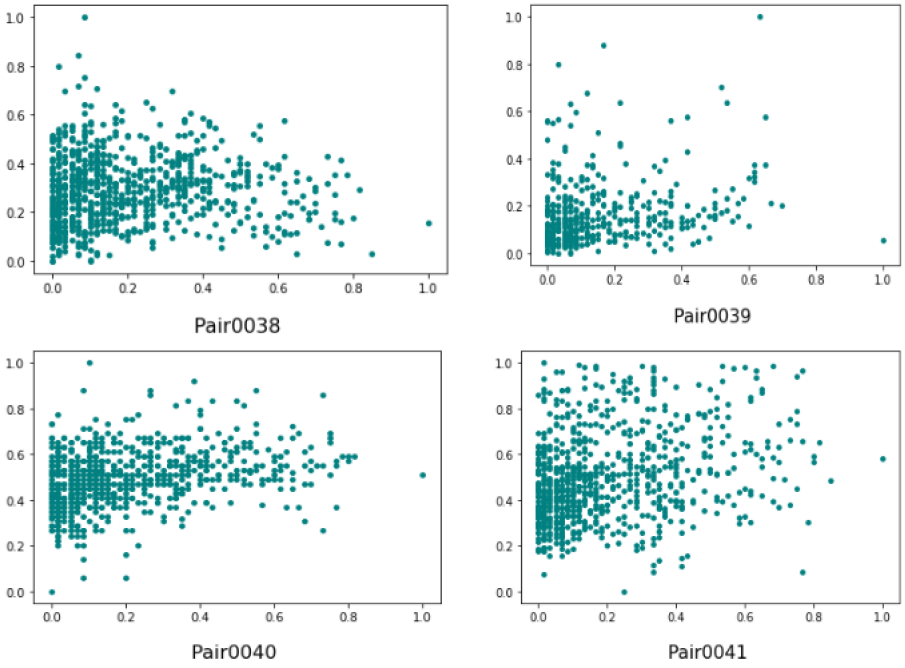

Figure 4.3.: Overview of scatter plots of pairs 38, 39, 40, 41 from the real Tübingen dataset; The values of variable x are represented by the x-axis, and the values of variable y are represented by the y-axis.

pair0039: Age –> Serum Insulin

2-hour serum insulin (μU/ml) is measured two hours after consuming a standard glucose dose during an oral glucose tolerance test. We can rule out the possibility that serum insulin causes age. There may be an effect of age on serum insulin, although another plausible explanation for the observed dependence could be selection bias [Moo+16].

pair0040: Age –> Diastolic Blood Pressure

Diastolic blood pressure (in mm Hg) is measured. It is clear that blood pressure does not cause age. While the reverse causal relationship appears plausible, selection bias could also provide an alternative explanation for the observed dependence [Moo+16].

pair0041: Age –> Plasma Glucose Concentration

Plasma glucose concentration is measured two hours after the ingestion of a standard glucose dose during an oral glucose tolerance test. Following similar reasoning as previously, we do not consider plasma glucose concentration to be a cause of age. However,

the reverse causal relationship seems plausible, and selection bias could also account for the observed dependence [Moo+16].

In section 7.3, we will demonstrate the performance of our proposed algorithm in inferring causal direction, comparing its results with those obtained from the state-of-art method like LOCI [Imm+22] and ROCHE [Tra+24]. Furthermore, we will investigate these pairs to illustrate how the MML score of our algorithm infer the causal direction.

5. Methodology

The CLS-MML algorithm is introduced in this chapter, along with the techniques employed and a detailed methodology of the experiment. It provides an overview of how the theoretical foundations are applied to infer causality and explains the metrics used to evaluate the causal methods in our approach.

5.1. Algorithm CLS-MML

The method CLS-MML to infer causality in a bivariate setting was introduced in [HM24]. It comprises of four main algorithms. Algorithm 1 and 2 employs the EM algorithm to estimate model parameters and their code lengths. Algorithm 3 determines the code lengths of the marginal distributions for the potential cause (X) and effect (Y). Algorithm 4 calculates a causal score for a given direction ($X \to Y$ or $Y \to X$) by combining Algorithm 1 or 2 plus 4. Finally, Algorithm 5 compares the scores for both directions to infer the estimated causal direction. This framework provides a structured approach to causality detection by leveraging model complexity and data structure.

X, Y normalized and are given by (x_i, y_i), $i = 1, \ldots, n$, assuming that Y follows lst distribution;

1. Standardize values of Y to follow Student-t
2. Find $\hat{\mu}, \hat{\tau}, \hat{\nu}$ by statistical fitting of y to Student-t as an initialization;
3. Compute $\hat{\beta}_{ML}$ as initial values;
4. Compute $\hat{K}$ from Eq. (3.2);
5. Compute $SNR = n\frac{\hat{K}}{\hat{\tau}}$;
6. For each $i = 1, \ldots, n$ compute the weights $\hat{w}_i$ from Eq. (3.4);
7. Compute $\hat{\beta}_0, \hat{\beta}$ from Eq. (3.5);
8. If SNR is high, compute $\hat{\tau}$ from Eq. (3.12) and go to step 10.
9. If SNR is low, compute $\hat{\tau}$ from Eq. (3.13) and go to step 10.
10. Otherwise compute $\hat{\tau}$ from Eq. (3.6).
11. Iterate steps 6-10 until convergence of $l(\boldsymbol{\theta})$ from (3.3).

Return $l(\hat{\boldsymbol{\theta}})$ with $\hat{\boldsymbol{\theta}} = (\hat{\beta}_0, \hat{\beta}, \hat{\tau})$ which is the minimum of (3.3).;

Algorithm 1: Computation of $l(\hat{\boldsymbol{\theta}})$

ALGORITHM 1 [HM24] presents this algorithm that estimates parameters in a regression setting where the dependent variable Y is assumed to follow a Student-t distribution. Steps 6 - 10 correspond to the EM algorithm that consists of several iterative and

sequential steps, combining statistical initialization and optimization to compute the optimal parameters $\hat{\beta}_0$, $\hat{\beta}$, and $\hat{\tau}$. Additionally, algorithm 1 will return the code length of estimation of the parameters of the regression model.

Normalization and Standardization : In the given data (X, Y) each observation pair (x_i, y_i) for $i = 1, \ldots, n$ is normalized, rescaling to the range [0, 1] and then target variable Y is standardized by transforming it to have zero mean and a variance of one.

Initial Parameter Estimation: Parameters $\hat{\mu}$, $\hat{\tau}$, and $\hat{\nu}$ of the Student-t distributions are estimated using statistical fitting methods. The initial estimates of $\hat{\beta}_{ML}$ are similarly estimated using the Maximum Likelihood Estimates (MLE) with a Student-t distribution.

Computation of $\hat{K}$: The value $\hat{K}$ is calculated by Eq. (3.2). This calculation is important to be able to compute the signal-to-noise ratio .

SNR Calculation: Algorithm 1 then proceeds to compute fixed value of SNR from the formula:

$$\text{SNR} = n\frac{\hat{K}}{\hat{\tau}} \tag{5.1}$$

Weight Computation: For each observation $i = 1, \ldots, n$, weights $\hat{w}_i$ are calculated using a designated formula (referenced as Eq. (3.4)). These weights are essential for the weighted regression analysis. This is the E-step of the EM algorithm

Weighted Regression: The algorithm updates $\hat{\beta}_0$ and $\hat{\beta}$ using a weighted regression formula (Eq. (3.5)). This is the first part of the M-step of the EM algorithm.

Estimation of Scale Parameter $\hat{\tau}$ Based on SNR:

If the SNR is high, $\hat{\tau}$ is computed using a specific equation for high SNR cases (Eq. (3.12)). If the SNR is low, $\hat{\tau}$ is computed using a different formula for low SNR cases (Eq. (3.13)). If the SNR does not fall into either conditions, $\hat{\tau}$ is computed using a general approach (Eq. (3.6)). We will discuss the differentiation between what is considered low and high in practice in section 5.4.2 . This is the second part of the M-step for the EM algorithm.

Iteration Until Convergence: Steps 6 to 10 are repeated iteratively until the objective function $l(\boldsymbol{\theta})$, defined without constraints, (referenced as Eq. (3.3)), converges.

Return Output: The algorithm returns the minimized value $l(\hat{\boldsymbol{\theta}})$ which is the code length of the optimal computed parameters along with the parameters $\hat{\theta} = (\hat{\beta}_0, \hat{\beta}, \hat{\tau})$.

ALGORITHM 2 As an alternative of algorithm 1, [HM24] proposes the algorithm 2. This approach incorporates a constraint for β values given by Eq. 3.2 based on the Signal-to-Noise Ratio to adjust the calculation of $(\hat{\tau})$ in different scenarios (high, low, or otherwise). In contrast, otherwise it does not account for SNR and updates $(\hat{\tau})$ based on the general equation.

Initialization: The initialization of parameters follows the same procedure as in the first algorithm. However, additional constraints are imposed on the parameters v and τ. The hyperparameter ν is bounded within limits that are adjusted based on the optimal results achievable by the algorithm for each specific dataset.

Main Loop (E-step and M-step): The algorithm then enters an iterative process, continuing until convergence of the log-likelihood function $l(\hat{\boldsymbol{\theta}})$:

Input: X, Y given by (x_i, y_i), $i = 1, \ldots, n$, and X, Y standardized.
Output: $l(\hat{\boldsymbol{\theta}})$ and $\hat{\boldsymbol{\theta}} = (\hat{\beta}_0, \hat{\beta}, \hat{\tau})$.
Compute $\hat{\nu}, \hat{\mu}, \hat{\tau}$ by fitting y to a Student-t distribution for initialization;
Compute $\hat{\beta}_{0ML}$ and $\hat{\beta}_{ML}$ by MLE as initial values;
Compute $\hat{K}$ from Eq. (3.2);
repeat
  foreach $i = 1, \ldots, n$ **do**
    Compute weights $\hat{w}_i$ from Eq. (3.4) ; `/* E-step */`
  end
  Plug $\hat{\beta}_0$ and $\hat{\beta}$ into $l(\boldsymbol{\theta})$ and minimize it with respect to $\hat{\tau}$; `/* M-step */`
until *convergence of* $l(\hat{\boldsymbol{\theta}}) = \min l(\boldsymbol{\theta})$ *from Eq. (3.3) with constraint* $\hat{K} - 1, \hat{\beta} \leq \hat{K} - 1$;
return $l(\hat{\boldsymbol{\theta}})$

Algorithm 2: Computation of $l(\hat{\boldsymbol{\theta}})$with constraints on $\hat{\beta}$

- **E-step:** For each data point $i = 1, \ldots, n$, the algorithm computes the weights $\hat{w}_i$ using Equation 3.4, and this step captures the expectation of the unobserved latent variables in the model.
- **M-step:** Once the $\hat{\beta}_0$ and $\hat{\beta}$ are computed using a weighted regression formula (Eq. (3.5)) considering this constraint: $\hat{K} - 1, \hat{\beta} \leq \hat{K} - 1$, the algorithm minimizes the log-likelihood function $l(\boldsymbol{\theta})$ with respect to $\hat{\tau}$.

The process repeats until the log-likelihood function $l(\hat{\boldsymbol{\theta}})$ converges, and the final values of the parameters $\hat{\beta}_0$, $\hat{\beta}$, and $\hat{\tau}$ are returned as the estimates that minimize the log-likelihood function along with code length of the obtained parameters.

Input: X, Y given by (x_i, y_i), $i = 1, \ldots, n$ and normalized
Output: $L(X), L(Y)$
Compute $\eta_X := \min\{|x_i - x_{i-1}|,\ 1 < i < n\}$ and
$\eta_Y := \min\{|y_i - y_{i-1}|,\ 1 < i < n\}$;
Compute $L(X) := -n \log \eta_X$ and $L(Y) := -n \log \eta_Y$;
return $L(X), L(Y)$

Algorithm 3: Computation of for $L(X)$ and $L(Y)$

ALGORITHM 3 [MV17]suggests this approach to calculate the code lengths $L(X)$ and $L(Y)$ for two normalized datasets X and Y, representing the cost of encoding the data based on the smallest differences between two data values.

1. **Input:** The algorithm takes two normalized datasets $X = \{x_i\}$ and $Y = \{y_i\}$, with $i = 1, \ldots, n$.
2. **Compute η_X and η_Y:**
   - $\eta_X := \min\{|x_i - x_{i-1}|,\ 2 \leq i \leq n\}$: Finds the smallest difference between elements in X.
   - $\eta_Y := \min\{|y_i - y_{i-1}|,\ 2 \leq i \leq n\}$: Finds the smallest difference for Y.
3. **Compute Code Lengths:** The code lengths are computed as:

$$L(X) := -n \log \eta_X, \quad L(Y) := -n \log \eta_Y$$

   These values represent the amount of information needed to encode X and Y based on their smallest resolution.
4. **Output:** Returns $L(X)$ and $L(Y)$.

---

1. Compute $L(X), L(Y)$ by Algorithm 3.
2. Compute $l(\hat{\boldsymbol{\theta}})$ by Algorithm 2.
3. Compute $\hat{\Delta}_{X\to Y} = (L(X) + l(\hat{\boldsymbol{\theta}}))/(L(X) + L(Y))$.
4. **Return** $\hat{\Delta}_{X\to Y}$.

Algorithm 4: Compute code for $\hat{\Delta}_{X\to Y}$

ALGORITHM 4 [HM24] computes the metric $\hat{\Delta}_{X\to Y}$, which quantifies the directional relationship between X and Y using code lengths from Algorithm 3 and code length of the estimated parameters from Algorithm 2 or Algorithm 1.

---

1. Compute $\hat{\Delta}_{X\to Y}$ and $\hat{\Delta}_{Y\to X}$ from Algorithm 4.
2. **If** $\hat{\Delta}_{X\to Y} < \hat{\Delta}_{Y\to X}$ **then** *causal direction* $:= X \to Y$,
else if $\hat{\Delta}_{X\to Y} > \hat{\Delta}_{Y\to X}$ *causal direction* $:= Y \to X$
otherwise *causal direction = undecided.*
3. **Return** *causal direction.* ;

Algorithm 5: Decide causal direction

ALGORITHM 5 By comparing $\hat{\Delta}_{X\to Y}$ and $\hat{\Delta}_{Y\to X}$, this method determines the causal direction or concludes that it cannot be decided.

All algorithms described above establish a robust framework for inferring causal directions in bivariate datasets. By combining statistical modeling, optimization techniques, and causal score comparisons, the approach offers interpretable insights, though the reliability of the results may differ according to dataset.

5.2. Overview of the Approach

The objective of this section is to describe and explain the steps and techniques which we employed to estimate the model parameters in the MML criterion and to infer causal relationships between variables using a Student's t-distribution-based regression model. This methodology integrates several key components: data preprocessing, parameter estimation, causal inference, constrained optimization, and tools and libraries.

5.2.1. Data Preprocessing

Data preprocessing is a crucial step to ensure that the data is scaled appropriately before applying the model. The input data (X and Y) is first normalized and target variable Y is standardized. The normalization scales data to a specific range: $[0, 1]$. Standardization ensures that all values have zero mean and unit variance. These steps are important because they improve the performance of the algorithm by ensuring features have similar ranges to each other. We do not give a feature too much weight just because its values are higher than those of other features. In Student's t-distribution, not scaling of input points can disproportionately influence the scale parameter (τ) resulting in parameter estimates that be skewed or biased. Furthermore, our model involves minimizing log-likelihood functions and not scaling can lead to poorly scaled gradients and for optimization algorithm such as Nelder-Mead [NM65] results may converge more slowly and be erroneous. There are pairs in Tübingen datasets that vary widely in scale. For example, weight, measured in pounds (lbs) or kilograms (kg), and fuel consumption, often measured in miles per gallon (mpg) or liters per 100 kilometers (L/100km), could be seen as two different variables. These two have very different ranges of values, the weight can have values in the thousands (a car may be 3,500 pounds), while fuel consumption usually has values in the range of single digits (10-30 mpg). In this case, weight can dominate the regression model creating some disproportion, which is challenging to determine how much fuel consumption affects the result in comparison to weight. Since our model is a Student's t-regression, we account for the potential presence of outliers and heteroscedasticity, concretely large values of a variable like weight could amplify the impact of outliers or noisy data, such as an very heavy car, might distort the regression model and conceal the true relationship of weight and fuel consumption. On the other hand, the trade-off is that during this process, the original values of the variables are lost. Despite this loss of interpretability, normalization and standardization are important because they ensure that the regression model can effectively identify the causal relationship between variables and the results are not biased by the scale of the data.

5.2.2. Initial Parameter Estimation

Fitting the Student-t Distribution The first step in the modeling process is to estimate the initial parameters for the Student's t-distribution. We use the `fit` method from the `scipy.stats.t` module that takes the effect (Y) as input to estimate the parameters location (μ), scale (τ), and degrees of freedom (ν) of the distribution. These initial

estimates act as the starting points for further optimization of our regression model. This approach ensures that the Student's t-distribution accurately reflects the characteristics of the data, especially under conditions of heteroscedasticity. However, one drawback is that if the data does not follow this type of distribution, the fitting may introduce extra complexity and computational overhead.

Maximum Likelihood Estimation We estimate $\hat{\beta}_{ML}$ (slope) and $\hat{\beta}_{0ML}$ (intercept) using MLE by maximizing the log-likelihood function of a Student's t-distribution. The optimization minimizes the negative log-likelihood, defined as

$$-\sum_{i=1}^{n} \log f_t(y_i|\boldsymbol{\theta}). \tag{5.2}$$

A numerical optimization method, `scipy.optimize.minimize`, was used to perform this computation. The negative log-likelihood is minimized using the `L-BFGS-B` optimization algorithm [Byr+95], a quasi-Newton method that prevents overfitting and supports bounds on parameters, ensuring reasonable constraints on the estimates. The function iteratively adjusts the parameters to minimize the objective, with the likelihood incorporating the predicted values $y_{\text{pred}} = \beta_0 + \beta x$ and accounting for the scale (τ) and degrees of freedom (ν). Constraints on the parameters were implemented as bounds, which were adjusted based on the characteristics of each dataset. A more detailed explanation of these adjustments is provided in the next section.

5.2.3. Parameter Bound Selection

In optimization processes, especially when we use the methods like L-BFGS-B, adding bounds to the feasibility of parameters is essential so the model estimation is effective and remain stable. The intervals for each parameter are chosen based on a combination of theoretical considerations, prior knowledge and empirical testing. In the set of Tübingen pairs, a lower bound of 10 or 8 is used for ν to prevent the model from overly weighting outliers and to ensure stable estimation of the distribution's tail. The upper bound for ν is typically either unconstrained or set to a large value such as 1000, allowing the distribution to approximate a normal distribution when ν is large.

For τ (scale), a lower bound of 1×10^{-6} is used to avoid numerical instability. This ensures the scale is never zero or too small, which could lead to extreme values for the model predictions. The standardization and normalization of the range of variables X and Y, both β (the slope) and β_0 (the intercept) have constant bounds $[-1, 1]$. Since our data is standardized (e.g., centered), setting this makes sense as 1-unit change in cause (x) leads to a 1-unit change in effect (y), which could be reasonable as our variables are on similar scales. This can act as a regularization technique, preventing the model from fitting extreme values in the data. On the other hand, placing these bounds over the parameters may restrict the model's flexibility and lead to biased estimates if the true parameter values fall outside this range. Considering these potential disadvantages, we also conduct experiments with unbounded parameters and present all results in the experiments chapter 5.4.

5.2.4. Constrained Optimization

The optimization of the objective function for Eq. (3.3 is subject to constraints on the parameter β (slope). Specifically, we impose the constraint :

$$-K - 1 \leq \beta \leq K + 1 \tag{5.3}$$

where K is a constant derived from Eq. 3.2. The value of K dynamically adjusts to the dataset through $\hat{K}$, reflecting the structure and variability of the input data. This constraint ensures that the estimated β values remain within a reasonable range, preventing the model to select values of β which would lead to overfitting.

5.2.5. Expectation-Maximization Algorithm

The EM algorithm [DLR77] is a widely-used iterative method for finding maximum likelihood estimates of parameters in models involving latent variables. We use the EM algorithm (as suggested by [WMS18] who employed it for the multivariate robust regression) to iteratively estimate the parameters β_0 (intercept), β (slope), τ (scale), and ν (degrees of freedom) in the presence of latent variables. This approach alternates between two steps: the E-step and the M-step.

In the E-step, we compute $\hat{w}_i$, the weights, which are based on the conditional expectations involving the degrees of freedom (ν), standardized response values, and current estimates of β_0, β, and τ. The weights ($\hat{w}_i$) allow the model to assign more importance to data points with lower noise, thereby improving parameter estimation (refer to Eq. (3.4)).

In the M-step, the parameters β_0, β, and τ are updated by minimizing the objective functions derived from the negative log-likelihood and additional constraints, such as bounds on the parameters (refer to Eq. (3.5) and Eq. 3.3)). The trust-constr method [CGT00], utilized in Python's `scipy.optimize.minimize`, is a constrained optimization algorithm ideal for solving the M-step, where parameters β_0, β, and τ are computed. It works by iteratively solving a series of approximations to the objective function, balancing accuracy and computational efficiency, which makes it suitable for handling weighted residuals and likelihood functions. Additionally, it ensures that each iteration makes meaningful progress without diverging, even in complex or noisy problems. This method was chosen primarily because empirical experiments on the data have shown that it consistently yields better results than other method in term of accuracy. For example, on the Tübingen dataset, this method achieved 70% accuracy, while other methods resulted in 60%. For the SIM dataset, this method also performed best with 60% accuracy, while others only achieved 50% accuracy, respectively.

This iterative process repeated up to 100 iterations until convergence criteria are met, defined by changes in the parameter estimates β_0, β, and τ dropping below a specified tolerance (1e-6), or the maximum number of iterations is reached. The final output of the algorithm includes the minimized message length, as well as the estimates for β_0, β, and τ, which characterize the fitted regression model. The EM algorithm is particularly beneficial in our context because it effectively handles the heavy-tailed nature of the Student's t-distribution and the presence of latent variables, enabling robust parameter estimation

despite heteroscedasticity. It is flexible in incorporating constraints and in optimizing multiple parameters at once. One of the key disadvantages of the EM algorithm, which will be shown in the experimental results where our method does not perform optimally, is its tendency to converge to local minima depending on the initialization of the parameters. To address this, we impose constraints and bounds on the initialization of the parameters. All their implications will be discussed in detail in the result chapter 7.

5.2.6. Causal Inference

The causal direction between two variables X and Y is inferred by comparing the strength of the relationship in both directions from X to Y and from Y to X. We compute two causal score using their minimum description length, $\Delta X \to Y$ and $\Delta Y \to X$, which represent the score of each causal direction based on the data. The direction with the higher score is considered the inferred causal relationship. Our method does not explicitly perform traditional statistical hypothesis testing but it compares the computed causal scores and it decides in this way which is the more likely causal direction.

5.2.7. Scalability and Efficiency

The CLS-MML algorithm is designed to handle datasets of any sizes, with computational complexity driven by the size of the input data and the number of iterations required for convergence. Efficiency is optimized through the use of specialized numerical libraries like **SciPy** for parameter estimation and optimization. Any dataset can be efficiently adapted to be processed independently, allowing scalability on distributed environments. Through empirical evaluation, the algorithm has demonstrated convergence in a relatively small number of iterations compared to alternative methods, while processing datasets in a short amount of time. For instance, the Tübingen dataset (having 99 causal pairs) is evaluated within approximately 10 minutes, while the Net, Cha, Multi dataset (having 300 causal pairs, each) requires around 20 minutes. Other datasets of type SIM (having 100 causal pairs) are completed in about 15 minutes. This efficiency in both convergence and processing time is a superior aspect of our algorithm.

5.2.8. Tools and Libraries

The methodology presented in this thesis leverages several Python-based tools and libraries, each chosen for its efficiency in handling data processing, statistical modeling, optimization and visualization. Key libraries used include:

- **Python**: The primary programming language, offering a large ecosystem for numerical analysis and optimization for our framework.
- **NumPy**: NumPy enables efficient handling of large datasets and array operations.
- **SciPy**: An important library for optimization, including the Nelder-Mead algorithm (via `scipy.optimize.minimize`), which is employed to minimize the negative log-

likelihood function in the model. SciPy also facilitates statistical modeling, especially the use of the Student's t-distribution for residual errors through `scipy.stats.t`.

- **Pandas**: Pandas handles the normalization manipulation, loading, and preprocessing of datasets, as well as the extraction of relevant features for causal inference.
- **Matplotlib**: Used for visualizing model performance, Signal-to-Noise Ratio values, and causal inference results. Visualizations also include data representations like graphs and scatter matrices to assess the distribution and relationships within the data.
- **Statsmodels**: used to access statistical models such as linear regression.

These libraries are selected for their robustness and flexibility in handling the mathematical and optimization tasks required by our model.

5.3. Evaluation metrics

To evaluate the performance of our method across all datasets, we use two common metrics: the Area Under the Decision Rate Curve (AUDRC), as introduced in [Imm+22], and the Forced Decision metric for assessing accuracy defined in [Moo+16].

5.3.1. AUDRC

According to [Imm+22], AUDRC is recommended as it distinguishes between $X \to Y$ and $Y \to X$ pairs with equal accuracy. This method avoids making arbitrary distinctions between true positives and true negatives, which could results on an unbalanced dataset that are difficult to interpret[Imm+22]. Accuracy reflects the proportion of correctly inferred cause-effect relationships, while AUDRC assesses how well decision certainty correlates with accuracy. Certainty in our case is indicated by the difference between the MML scores in both directions. Therefore, a high AUDRC suggests that an estimator is typically accurate when it is confident and only makes errors when uncertain. Formally, given a causal estimator $f(\cdot)$ applied to M pairings, arranged by the estimator's certainty using $\pi(\cdot)$, and the ground truth direction $t(\cdot)$, [Imm+22] defines

$$\text{AUDRC} = \frac{1}{M} \sum_{m=1}^{M} 1_{f(\pi(i))=t(\pi(i))} \tag{5.4}$$

This means that we calculate the average accuracy by progressively adding the pairs for which the estimator is most confident. Consequently, AUDRC and accuracy both fall between 0 and a maximum of 1. A higher AUDRC indicates that the method successfully predicts the correct causal directions when the score confidence is high, while errors are associated with lower confidence in the predicted scores. This method is used to evaluate all 13 datasets in the same way, ensuring consistency across all evaluations.

5.3.2. Forced Decision: Evaluation of accuracy

Given a pair of variables (X,Y), the methods must decide whether X→Y or Y→X, and we evaluate the accuracy of these decisions as follows according to [Moo+16]:

$$\text{Accuracy} = \frac{\sum_{m=1}^{M} w_m \delta_{\hat{d}_m, d_m}}{\sum_{m=1}^{M} w_m}$$

where the weight of the pair is w_m, the estimated direction is $\hat{d}_m$ (i.e., ←, →, or ? for undecided), and the true causal direction for the m-th Tübingen pair is d_m (either ← or →).

[Moo+16] noted that only correct decisions are awarded. Hence, if no estimate is provided (d_m =?), this will negatively impact the accuracy. This metric is particularly used for real datasets, where the pairs have different weights. The accuracy is determined by dividing the number of successfully detected pairs by the total number of pairs in the synthetic datasets, where no weight is assigned to any pair.

5.3.3. Weights

[Moo+16] further elaborates on the importance of weighting in the Tübingen dataset. When analyzing the cause-effect pairs, pairs from the same dataset cannot always be treated as independent. This is because some variables in the dataset may be correlated, which can undermine the assumption of their independence. [Moo+16] gives a example that there is a strong relationship between variables such as shucked weight, viscera weight, whole weight, and shell weight in the Abalone dataset. However, treating pairs like (age, full weight), etc., as independent could generate bias, because of their underlying relationships [Moo+16]. To address this issue, [Moo+16] correct for this bias by down weighting these correlated pairs. The goal is to adjust the weights in a way that accounts for these dependencies. Specifically, the weights are chosen so that the total weight for all pairs from the same dataset equals one, and the weights for each pair are equal. For Tübingen cause-effect pairs, the weights are given in table 4 in [Moo+16]. In contrast, for the other artificially simulated cause-effect pairs, no weighting is applied because the datasets variables are assumed to be independent in this case.

5.4. Experiments

5.4.1. Setup

The evaluation of the CLS-MML algorithm was conducted using 13 datasets which we described in Section 4, each processed individually to ensure accurate extraction of causal pairs and associated ground truth labels. Each dataset was read separately, allowing the causal pairs (x, y) and their corresponding ground truth causal directions to be prepared for analysis. Variations in file structure were handled programmatically, ensuring robust parsing. For instance, values were accurately extracted from comma-separated formats,

and metadata lines were skipped. This allowed the ground truth causal direction labels to be aligned with the extracted pairs for evaluation.

For all experimental setups, each dataset was processed independently, enabling flexibility in running the CLS-MML algorithm on specific datasets or subsets. All steps, from data parsing and preprocessing to running the experiments and calculating metrics, were automated via scripts. For instance, files in a format such as `ANLSMN_pairs/MN-U` were processed to extract x, y, and their associated ground truth directions, while functions like `estimated_direction` calculated causal inferences. Evaluation metrics were derived via code, ensuring a reproducible and efficient pipeline for all experiments. Our code is available at https://github.com/anonymgitH/clsmml, providing full access to the implementation for reproducibility and further experimentation.

The experiments were separated into two distinct groups:

1. The experiments based on evaluating the algorithm's performance under different signal-to-noise ratio conditions.
2. The experiments focused on optimizing the algorithm using initial parameter estimates to assess its ability to infer causal relationships starting from approximate values.

This division allowed for a comprehensive analysis of the algorithm's robustness and its performance with different data characteristics.

5.4.2. Experiment 1

In the first experiment, we focus on evaluating the performance of the EM algorithm based on Signal-to-Noise Ratio thresholds (see algorithm 1). The objective is to observe how different SNR splitting approaches influence the calculation of the τ parameter and consequently the accuracy of the causal directions for each dataset. To achieve this, we applied several threshold methods for classifying SNR as low or high, based on the dataset-specific characteristics. These thresholds were set according to various strategies, such as ScatterPlot-based distribution analysis, statistical properties of input data (x), values derived from calculated $\hat{K}$ and other custom rules. Table 5.1 summarizes these thresholding strategies, detailing the corresponding low and high SNR thresholds, how the τ parameter was computed and the methodology used to estimate the degrees of freedom (ν) in each case. The outcomes of these experiments will be presented in Chapter 7, where we will analyze the impact of these various SNR thresholds on the performance metrics and parameter estimations.

Alternative SNR Definitions

The Signal-to-Noise Ratio serves as an important metric for assessing data quality and its impact on parameter estimation within the algorithm. There is no unique definition of SNR in the literature, see e.g. [Wik24a]. Initially, we adopted the SNR definition from [WMS18], expressed in Eq. 5.1 where K denotes the signal strength, n is the data set size

Approach	SNR Low Threshold	SNR High Threshold	τ Calculation	ν Value	Notes
ScatterPlot-Based	Determined from the lower values of the SNR Scatter-Plot for one dataset.	Determined from the upper values of the SNR Scatter-Plot for one dataset.	Computed based on the high equation 3.12, low equation 3.13 or general equation 3.6.	Estimated as part of the parameter initialization.	Utilized empirical distribution to define boundaries.
Mean and Standard Deviation of x	$(\text{Mean}(x) - 8 \cdot \text{Std}(x))$	$\text{Mean}(x) + 2.5 \cdot \text{Std}(x)$	Set using simplified rules for low, high and moderate SNR cases.	Estimated as part of the parameter initialization.	Adjusts thresholds based on statistical properties of x.
Using $\hat{K}$	Based on values derived from $\hat{K}$ calculations (e.g., 20% below the mean).	Based on values derived from $\hat{K}$ calculations (e.g., 20% above the mean).	Set using simplified rules for low and high SNR cases.	Estimated as part of the parameter initialization.	Linked to scale of $\hat{K}$ to reflect dataset-specific characteristics.
Fixed General Rule	A fixed value of 1 for all datasets	A fixed value of 10 for all datasets	Set using simplified rules for low and high SNR cases.	Fixed value during initial fitting of Student's t-distribution for consistency.	Uniform thresholds applied across datasets.
Using τ **and** μ	Dynamic estimation of μ - 2 τ	Dynamic estimation of μ + 2 τ	Optimized dynamically based on EM-algorithm and current τ, μ estimates	Dynamic optimization using current τ	Dynamic optimization based on previous estimates and minimization process

Table 5.1.: Thresholding approaches for SNR; By Mean(x), we refer to $\hat{\mu}$; By Std(x), we refer to $\hat{\tau}$; By $\hat{K}$, we refer to the Eq. (3.2)

and τ is the scale parameter. To examine the sensitivity of the algorithm performance to the SNR formulation, we experimented with alternative definitions. One approach was to define the SNR alternatively based on the statistical properties of the input data,

$$\text{SNR} = \frac{\text{mean}(x)}{\text{std}(x)}, \tag{5.5}$$

where $std(x)$ is the standard deviation of the signal x, emphasizing the characteristics of the signal. Additionally, we explored another more general SNR definition from Eq. 3.8. These variations allowed us to investigate how the choice of SNR affects parameter stability, EM convergence, and overall algorithm performance.

5.4.3. Experiment 2

In this second experiment, the SNR split was removed, and we focused on investigating the impact of parameter initialization methods before the EM algorithm. Specifically, this experiment considers two main scenarios: assuming Gaussian errors and assuming Student's t-distribution errors for the regression parameters. Additionally, the degrees of freedom (ν) and the scale parameter (τ) are either fixed from the initial Student's t-distribution fitting or estimated using the maximum likelihood estimate. During MLE, we enforce constraints or boundaries on the parameters β, β_0, ν, and τ to ensure that the parameter estimates remain computationally stable. Table 5.2 summarizes the different experiments and conditions considered. This experiment allows us to evaluate the effect of different error assumptions and parameter initialization methods on the performance of the algorithm, while ensuring that the estimates remain stable and within expected ranges.

Experiment	**Error Assumption**	**Fixed of MLE**	**Notes**
1	Gaussian error for β and β_0	Fixed ν and τ from Student's t fitting	Initial values based on Student's t-distribution fitting, with constraints on parameters.
2	Gaussian error for β and β_0	MLE for ν and τ	ν and τ estimated from data using MLE, with parameter constraints.
3	Student's t-distribution error for β and β_0	Fixed ν and τ from Student's t	Initial values based on Student's t-distribution fitting, with constraints on parameters.
4	Student's t-distribution error for β and β_0	MLE for ν and τ	ν and τ estimated from data using MLE, with parameter constraints.

Table 5.2.: Comparison of different initializations for parameters and error assumptions with constraints on the parameters.

6. Related Work

This section provides a thorough review of the research conducted on the cause-effect bivariate inference within the context of the heteroscedastic and homoscedastic noise model. The literature can be categorized into two main groups: one consisting of methods based on cause-effect asymmetry, and the other based on algorithmic independence between the cause and the mechanism.

6.1. Methods based on cause-effect asymmetry

Many well-known techniques in the bivariate context are those based on the idea of cause-and-effect asymmetry. Build in the concept that, under certain assumptions the causal relationship between variables can only go in one direction and not the reverse, researchers have studied methods to identify the true causal models.

It has been demonstrated that the backward model does not exist for many additive noise models. The earliest identifiability results were established for linear additive noise models [Shi+06], where the effect Y is represented as a linear function of the cause X with an additive non-Gaussian error term. Many researchers have followed this direction for non-linear [Hoy+08], and post-nonlinear relationships [ZH12] as well.

Depending on whether the model assumptions are met, methods for assessing the score asymmetry of causal inference for various linear and non-linear SCMs are suggested for both directions. Common scoring methods include the log-likelihood and non-parametric independence test scores, such as the Hilbert-Schmidt Independence Criterion (HSIC) [Gre+05], or the mutual information between the residual and the cause.

For homoscedastic noise models, where the independent variable is assumed to influence only the mean and the variance of the dependent variable remains unchanged, many methods use regression techniques to fit these models [Shi+06], [Pet+14], [BPE14]. CAM [BPE14] method performs causal inference by decoupling order search from feature selection in high-dimensional additive structural equation models, using maximum likelihood estimation for order search and sparse regression techniques for edge selection to efficiently identify the causal structure. RESIT [Pet+14] method explores learning causal DAGs from observational data, showing that if the data follows a structural equation model with additive noise, the DAGs can be uniquely identified, followed by an additional step of independence scoring.

For heteroscedastic noise models we have methods like HECI [Xu+22], LOCI [Imm+22] and ROCHE [Tra+24]. HECI [Xu+22] identify causal direction between two continuous variables by modeling heteroscedastic noise through partitioning the cause's domain and optimizing a penalized regression score. It uses the Bayesian Information Criterion

(BIC) to regularize the size of the partition and the complexity of the fitted functions. LOCI [Imm+22] method examines location-scale orLSNM, where the effect Y is modeled as a function of the cause X with noise scaled by a positive function of X. It presents two estimators for maximizing the Gaussian log-likelihood: one based on feature maps, ensuring joint concavity and consistency, and another using neural networks capable of modeling complex functions. The method utilizes the log-likelihood from the mean and variance parameters of the Gaussian distribution as a criterion, and then recovers the residual for testing independence with the Hilbert-Schmidt Independence Criterion (HSIC). Additionally, method ROCHE follows a procedure similar to that of LOCI. Since Loci uses a the Gaussian likelihood as the optimization objective which can not be optimal when the data follows a non-Gaussian distribution, Roche comes with a novel approach that estimates the model using Student's t-distribution and leverages neural networks. This modification can be more robust in handling variability and extreme values maintaining the distribution's general form. ROCHE and LOCI are currently the state-of-the-art methods for determining causality between two variables in bivariate setting.

6.2. Methods based on independence between cause and mechanism

The second category of methods is based on the **Postulate 1** and **Postulate 2** of independence between the cause and the mechanism, where our method falls into this category. This principle is not just a theoretical assumption (a postulate) but also can be used to identify the true causal structure of SCM.

In location-scale models, the noise can be connected to both the independent and dependent variables, making it more difficult to determine causal direction than in simple additive noise models, where the noise does not depend on the independent variable. Some methods try a direct approximation of distributions [Jan+12] [Sgo+15] [Gou+18], or adopt a information-theoretic approach such as comparing the Kolmogorov complexities of different factorizations in both directions [JS10].

Method Information Geometric Causal Inference (IGCI) from [Jan+12] extends traditional causal inference by considering the shape of conditional distributions, assuming that the causal hypothesis "X causes Y" reflects independent mechanisms. IGCI defines independence via orthogonality in information space, capturing asymmetries between cause and effect (which are deterministically related), making it particularly effective in cases with low noise levels.
Method Statistical Low-complexity Penalization (SLOPE) from [MV17] uses an information-theoretic approach based on Kolmogorov complexity and the Minimum Description Length principle. The proposed method uses MDL-based regression to determine causality y comparison Y's description lengths with X's and vice versa, selecting the direction with the shorter description. SLOPE is a linear-time algorithm shown to significantly outperform existing methods on both synthetic and real-world data.

Inspired by the invariance of the causal mechanism, Conditional Divergence-based

Causal Inference (CDCI) [DN22] estimates the stability of the conditional distribution and uses its inverse, the divergence, to detect causal relationships. This approach relaxes strict assumptions commonly used in causal discovery, such as specific functional forms or noise models, and supports arbitrary measures of distribution divergence.

Lastly, method Bivariate Quantile Causal Discovery (QCCD) from [TCV20] uses the Minimum Description Length principle linked to quantile regression to infer causal direction from observational data. By leveraging multiple quantile levels, QQCD adapts to various generating mechanisms, including additive, multiplicative, and location-scale models.

7. Results

This section presents the results of all experiments conducted to evaluate the performance of CLS-MML algorithm under different experimental setups. Key insights are drawn from the parameter optimization, SNR splitting strategies, and their impact on the accuracy of the causal scores.

7.1. SNR Threshold Results

7.1.1. ScatterPlot-Based SNR Threshold Selection

In this experiment, we test Algorithm 1 where we derive the SNR thresholds from the scatter plot of SNR values calculated for each dataset. This method allowed the dynamic choice of low and high thresholds, based on the density distribution of SNR values, providing a dataset-specific partitioning of the SNR range. This approach is consistent with the heuristic methods often used in statistical threshold techniques. The point at which the signal is much stronger than the noise or at which the noise has little to no impact on the model's performance are common practical way that are frequently used to create thresholds for low and high SNR in statistical modeling and related fields.

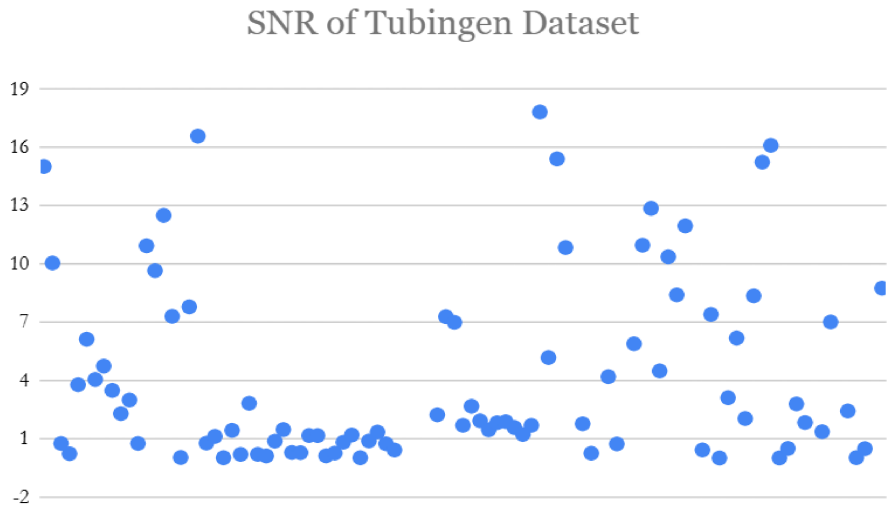

Figure 7.1.: Scatter plot of SNR values for all 99 Tübingen pairs: X-axis denotes the number of a Tü pair, Y-axis the SNR value.

The scatter plot analysis of Figure 7.1 shows that SNR values of Tübingen pairs clustered around certain range. The majority of values are clustered within the range of 0.5 to 10 which we consider as moderate case of SNR. Any value below 0.5 is considered as low SNR, indicating situations where the signal is weak compared to the noise. Conversely, values above 10 are categorized as high SNR, where the signal dominates, and the influence of noise is minimal. This approach relies on empirical observation rather than being derived from a strict theoretical model. The selection of this threshold was made by visually inspecting the scatter plot and comparing the performance of the model with different ranges. We were able to find thresholds using this heuristic approach, which increased the model's precision in determining causal directions, achieving 67% accuracy in the best-case scenario, compared to 62% accuracy without threshold splitting. Although these thresholds are not derived from a strict statistical rule, these thresholds provide a good approach that create a balance between the model's performance and sensitivity to noise in real-world data.

On the other hand, since the Tübingen dataset contains pairs from various domains, including environmental data, biological data, and economic data, it is likely that the optimal threshold for each domain would vary. Thus it is hard to expect that one can find a universal method for selecting a SNR threshold assuring a high causal score on the whole Tübingen dataset. In general, real-world applications, it is more reasonable to expect that an expert knowledge can provide the right SNR threshold for specific dataset.

In the Abalone dataset, the pair Age vs. Length represents a biological relationship where Age is a predictor for Length of an Abalone shell. This kind of biological data is noisy, as biological measurements are influenced by a variety of factors (e.g., environmental conditions, measurement errors) that introduce variability into the data. If we use a threshold range of 0.5 to 10 for the SNR in this pair, we might mistakenly label the relationship as "low SNR" even if the connection between Age and Length is important, but noise. This could lead to the wrong conclusion that the link between Age and Length is weak or unclear, when in reality, the signal is just hard to detect because of the noise. Consequently, the model may be unable to accurately detect the causal relationship. This would lead to a loss of accuracy, as the model would by mistake treat this relationship as weak. As stated before, there is no definitive rule to set the threshold for SNR based only on scatter plots, and it may produce false results, if not properly adjusted. In summary, using scatter plots to determine SNR thresholds in the Tübingen dataset is a useful tool that helps understand the data's structure. However, having domain knowledge helps to avoid incorrect classification and improve the consistency of results.

Now let us consider SIM-ln, one of the synthetic dataset. The SIM-ln dataset, characterized by its low noise levels, is created to test causal inference models in situations where the signal is mostly strong. Its resulting SNR values span from 1 to 10, with a significant clustering around 9-10, indicating strong signal in most cases, while a smaller fraction falls between 1 and 8, representing scenarios where noise has a effect. For example, thresholds can classify values below 5 as low SNR, meaning moderate noise; values between 5 and 9 as moderate SNR, where the signal is strong but some noise is present; and values above 9 as high SNR, where the signal is very clear. As stated before the chosen boundaries are

empirically motivated by observing the SNR distribution of the dataset and its effect on performance. Without the threshold, we achieved an accuracy of 52%, whereas with the threshold, the accuracy increased to 56%. This demonstrates that applying the threshold slightly improves the model's ability to distinguish between cases with varying noise levels.

The usefulness of thresholds for all datasets is demonstrated by even a small improvement (4-5%) over the baseline. This gain suggests the thresholds usage, capture key distinctions in the data. Without thresholds, all data pairs are treated the same, which can reduce the accuracy of causal models, especially for data with very low or very high SNR. While these thresholds are dataset-specific, this strategy for thresholding can be extended to other datasets as well.

But comparing Figure 7.1 and Figure 7.2, one can see that majority of the data pairs in SIM-ln have SNR values within the same interval, while the SNR values for Tübingen pairs are more diverse.

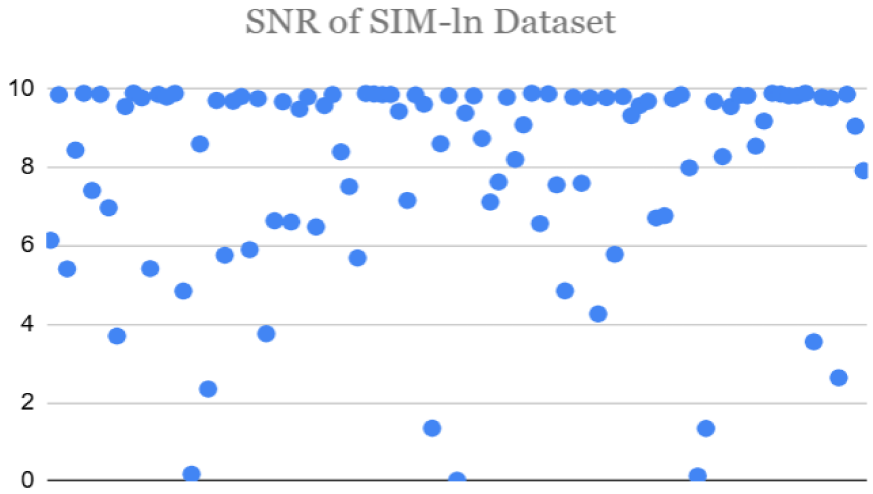

Figure 7.2.: Scatter plot of SNR values for SIM-ln dataset

7.1.2. Statistically-Based SNR Threshold Selection

This SNR thresholding strategy uses statistical properties of the dataset such as mean and standard deviation to classify the Signal-to-Noise Ratio into low, high, and moderate categories. The initial choice of using this approach is grounded in basic statistical principles. Initially, we started with the basic approach of using mean minus standard deviation for the low SNR threshold and mean plus standard deviation for the high SNR threshold. This provided a simple way to define the boundaries based on the central tendency and spread of the data. However, we then introduced a scaling factor to better refine these thresholds and capture the specific characteristics of the dataset. While the initial approach of using mean $\pm$ std works well for standard distributions, real-world

data often contains variations such as distribution skewness, or noise. We were able to fine-tune the low and high SNR limits thanks to the scaling factor, which guaranteed a more accurate data classification. By using as scale of -8 for the low threshold and $+2$ for the high threshold, we were able to better separate low and high SNR values, improving the accuracy of the causal direction estimation. While each dataset may require its own specific settings, the general approach of using these scaling factors led to noticeable improvements for most of the 12 datasets. Especially, the Tübingen dataset showed the best results, up to 72% and consequently the most accurate classification of causal directions (see Table 7.1).

	Tü	**AN**	**AN**	**LS**	**LS**	**MNU**	**SIM**	**SIM-ln**	**SIM-G**	**SIM-c**	**Multi**	**Net**	**Cha**
T	72	45	50	44	48	50	43	52	53	56	83	53	60
NT	62	44	48	43	47	50	41	50	52	55	81	53	60

Table 7.1.: Comparison of causal accuracy score for all datasets with thresholds set at low = Mean(x) - 8 * Std(x) and high = Mean(x) + 2.5 * Std(x); T denotes SNR threshold, NT denotes No SNR threshold

7.1.3. Dynamic Thresholding Based on μ and τ

In this approach, the SNR thresholds are determined dynamically using the estimates of μ (location parameter) and τ (scale parameter). The thresholds are defined as: lower threshold μ - 2τ and upper threshold: $\mu + 2\tau$. Unlike static thresholds based on predefined scaling factors, this method adjusts the thresholds during the optimization process. The values of μ and τ are updated iteratively within the EM algorithm framework, reflecting the current state of the parameter estimates. This ensures the thresholds adapt to the underlying data distribution at each step. By using optimized thresholds, we expect the model to achieve better threshold of SNR , leading to improved accuracy in estimating causal directions.

Let's consider again the causal pairs with the relationship between the age and length of abalones in the Tübingen dataset. In younger abalones, where age measurements are more variable, the thresholds constantly adapt to narrow the region of high SNR, avoiding the influence of noise. In contrast, for older abalones, where the relationship between age and length is more stable, the thresholds expand, capturing stronger signals. This strategy enhances the identification of length $\rightarrow$ age as the causal direction, going over static threshold methods, which lack the flexibility to address the heteroscedastic nature of the data. However, the accuracy of correctly detected causal directions did not exceed the statistical barrier, despite our expectation that it would increase with dynamic thresholding. For the Tübingen dataset, the accuracy of CLS-MML with this dynamic thresholding reached 70%, but for the rest of the datasets, the improvement was marginal, with accuracy only showing a 1-2% increase over the no-threshold approach. This implies that although the dynamic thresholding approach had some advantages in particular situations, like Tübingen, it did not consistently raise CLS-MML's causal direction accuracy in other synthetic datasets.

7.1.4. Uniform Thresholds for SNR

We aimed to have a general rule for the SNR values for all datasets, using a general rule for SNR classification where values less than 1 were considered as low, and values greater than 10 were classified as high, with anything in between being considered moderate. The results for the synthetic datasets showed slightly lower improvements in accuracy. For the Tübingen dataset, however, the improvement was very insignificant, with accuracy only increasing by about 2%, from 62% to 64%. This method provided a simple, uniform classification of SNR values across datasets, but the impact on model performance was minimal in many cases. The approach of using fixed SNR thresholds is not ideal for both real and synthetic datasets because it fails to adapt to the unique characteristics of each dataset. The lack of flexibility in the approach leads to minimal improvement in accuracy, as seen in Tübingen, where the increase was only 2%. Moreover, the fixed thresholds simplify the classification process too much, ignoring the detailed distribution of SNR values, which can lead to misclassify data and lowering our model accuracy.

7.2. Parameter Adjustment Results

In these experiments, we use Algorithm 2, where the SNR threshold is removed and parameter initialization is employed to achieve the highest accuracy in causal direction identification across all datasets.

Firstly, we computed the MLE by assuming a Gaussian errors in the robust regression model. This approach involved fitting the model to the data and estimating the parameters β and β_0. Additionally, for degree of freedom ν and scale parameter τ, we considered two scenarios: one where they were fixed based on the initial fitting and another where they were computed using MLE. We made this decision to see how the accuracy would be affected by the initial parameter estimation, particularly when assuming Gaussian errors. This approach allowed us to assess if simplifying the estimation process with Gaussianity could yield effective results, particularly when dealing with data that may not have extreme outliers or heavy tails. For example the pair pair0084 from the Tübingen dataset: Population $\rightarrow$ Employment, we initially assumed a Gaussian error distribution, which is a reasonable assumption in cases where the relationship between employment and population is consistent, and have no extreme outliers. Because the relationship is predictable, assuming Gaussian errors in our model, it correctly determined the direction of this variables. However, when used the Student's t-distribution, the model wrongly predicted the causal direction. In the case of a Student's t-distribution, it may occur that outliers or extreme events (such as large economic shocks or demographic anomalies) could heavily influence the parameter estimates, causing their causality to be incorrect. After conducting all experiments assuming a Gaussian error distribution, we obtained the results summarized in Table 7.2. This approach generally worked well for pairs where the relationships were free of significant outliers, such as Population $\rightarrow$ Employment, Gross National Income (GNI) per capita $\rightarrow$ Life expectancy and more, yielding correct causal directions. However, for pairs with heavy-tailed data, the assumption of Gaussianity may

have affected the accuracy, as outliers were not accounted for.

Secondly, we ran experiments to compute the MLE by assuming a Student's error for initial estimation of parameters, as we believe our datasets mostly contains heavy-tailed noise. Also we imposed specific constraints to maintain stability and align with the dataset characteristics. β was bounded initially between -1 and 1 and when optimized with the constrains K-1 and K+1 to reflect realistic causal strengths, while τ the scale parameter, was kept strictly positive with a lower bound of 10^{-6} to avoid instability. Additionally, ν representing the degrees of freedom in the Student's t-distribution, was treated as a hyperparameter, adjusted across datasets to account for diffrent levels of tail-heaviness in the data. For the Tübingen, Multi, CHA, and Net datasets, the hyper-parameter ν interval is set to [200, 1000]. For the AN, AN-s, LS, LS-s, and MN-U datasets, the interval is set to [100, infinity] and for the SIM, SIM-ln, SIM-c, and SIM-G datasets, the interval is set to [10, infinity]. These constraints ensured meaningful initialization and robust optimization. In Table 7.2 you can find the best results from the experiment with these adjustments. This shows how important it is to choose the right initialization for the data, as it can have a big impact on the results.

	Tü	**AN**	**AN**	**LS**	**LS**	**MNU**	**SIM**	**SIM-ln**	**SIM-G**	**SIM-c**	**Multi**	**Net**	**Cha**
GA	67	45	50	44	47	50	43	52	54	56	72	55	56
SA	70	45	55	51	61	46	46	49	54	61	74	60	58
NA	62	44	48	43	47	50	41	50	52	55	81	53	60

Table 7.2.: Comparison of parameter estimations across different assumptions. The table shows the results for accuracy of all datasets using three approaches: (NA) **No Adjustment**: No specific adjustments made to the parameters, (GA) **Gaussian Assumption of errors **: Parameters initialized assuming a Gaussian error distribution, and (SA) **Student-t Assumption of errors**: Parameters initialized assuming a Student-t error distribution. The results demonstrate the effect of each initialization approach on accuracy across all datasets.

7.3. Causality for Pima Indians Diabetes Dataset

Considering table 7.3 for the Pima Indians Diabetes pairs discussed earlier on section 4.2.1, our method CLS-MML correctly identified causality in 3 out of 4 cases, whereas Loci identified causality in only 2 out of 4 cases. However, Roche accurately identified causality in 3 out of 4 cases, similar to our results. It is important to point out that each of the four pairs carries equal importance in determining causality, with a weight of 0.25 assigned to each pair.

Dataset	**CLS-MML**	**Roche**	**Loci**
Pair 38	$\rightarrow$	$\rightarrow$	$\rightarrow$
Pair 39	$\rightarrow$	$\rightarrow$	$\rightarrow$
Pair 40	$\rightarrow$	$\leftarrow$	$\leftarrow$
Pair 41	$\leftarrow$	$\rightarrow$	$\leftarrow$
Ground Truth	$\rightarrow$	$\rightarrow$	$\rightarrow$

Table 7.3.: The table shows how our method CLS-MML performs compared to state-of-the-art LOCI [Imm+22] and ROCHE [Tra+24] on 4 real pairs of Tübingen dataset. The ground truth and the estimated causality direction for each pair are presented.

In the figure 7.3, we show how our CLS-MML algorithm performs for pairs 38 (Age $\rightarrow$ BMI) and 41 (Age $\rightarrow$ Plasma Glucose Concentration) of Pima Indians Diabetes data set. For pair 38, the algorithm correctly identifies the causal direction as $x \rightarrow y$, while for pair 41, it fails to determine the correct direction. Specifically, for pair 38, the MML code length is 0.3922 for the direction $x \rightarrow y$ and 0.5915 for the reverse direction $y \rightarrow x$. Since a shorter code length indicates a better fit, the algorithm selects the causal direction $x \rightarrow y$ because $\Delta_{x \rightarrow y} < \Delta_{y \rightarrow x}$. In contrast, for pair 41, the MML code length is 0.1482 for the direction $x \rightarrow y$ and -0.7971 for the direction $y \rightarrow x$, since $\Delta_{y \rightarrow x} < \Delta_{x \rightarrow y}$ the algorithm select the causal direction $y \rightarrow x$, which results in an incorrect estimation. Our CLS-MML algorithm correctly estimates the causal direction for pair 38 because the relationship is likely strong and aligns well with the Student's t-distribution error model assumptions. In contrast, pair 41 may involve weaker causal signals or confounding variables (e.g. diabetes status) that may lead to incorrect inference.

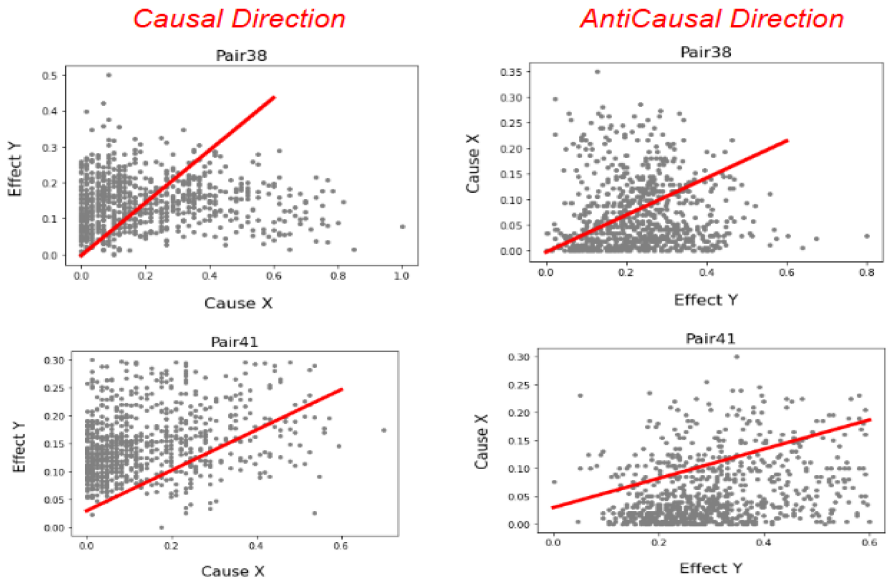

Figure 7.3.: Results of the CLS-MML method on pairs 38 and 41 from the Tübingen dataset for Pima Indians Diabetes. For pair 38, the x-axis represents age, and the y-axis represents body mass index. For pair 41, the x-axis represents age, and the y-axis represents plasma glucose concentration. The figures on the right show the causal direction $x \to y$, while the figures on the left show the reverse direction $y \to x$. The line passing through the figures represents the robust regression line obtained from the parameter estimation.

8. Evaluation and Discussion

In this section, we will discuss the results of the proposed CLS-MML method, evaluate its performance, and benchmark it against existing techniques in the field of bivariate causality. Additionally, we will highlight the contributions this work brings to advancing causal inference. Finally, we will outline potential directions for future work to further enhance and extend this research.

8.1. Interpretation of the results

As mentioned in the preceding sections, the numerous tests carried out using the suggested algorithm from [HM24] for bivariate causality show that the algorithm, backed by its theoretical foundation, provides a promising approach method for accurately determining causality on the real data set of Tübingen pairs and other synthetic datasets. Among all the tests, the approach that proved most effective was adjusting parameter initialization rather than using a SNR threshold adjustment. Classifying SNR as high, low, or moderate for a dataset can be problematic if the dataset lacks a clear distinction between the signal (the true underlying pattern) and the noise (random errors). SNR makes the assumption that the signal and noise components are distinct and well-defined, but this is not the case in our datasets. Moreover, SNR classifications are context-dependent, and using a generalized approach across different types of data sets may result in oversimplified or incorrect conclusions. Adjusting parameter initialization is often a better approach because it provides a more structured and flexible way to optimize the model. When using parameter initialization methods, like fitting the data to a Student's t-distribution and computing MLE for β, β, τ, ν, we were setting the starting point, i.e. initialization of the optimization process to values that are already statistically grounded and close to the optimal solution.

Combining the results of the previous chapters, we can make the following observations about the results we obtained:

- Initializing parameters through statistical fitting and MLE resulted in more stable and faster convergence during optimization and consequently to a higher score of correctly detected Tübingen pairs.
- Setting specific intervals for hyper-parameters based on the dataset type improved accuracy.
- Student-t parameter assumption approach generally provided a higher accuracy than Gaussian parameter Assumption), particularly for datasets like Multi and Net.

- SNR-based classification (high, low, moderate) was found to be less reliable for some datasets such as AN, ANs, LS, LSs and MNU.
- The effectiveness of the SNR threshold varies by dataset, with some datasets showing only marginal improvements.
- CLS-MML when applied on the Tübingen dataset outperformed the methods due to its real-world nature, which likely provides more complex and relevant patterns, whereas the synthetic datasets may lack the natural variability and authenticity needed for optimal model performance.

While the CLS-MML effectiveness varies among different data sets, none of them consistently achieve very high accuracy (above 80% or 90%), and in some cases, the results appear to be random. This suggests that while the methods can be effective, the parameters may not be fully optimized for all datasets. Improving model assumptions, fine-tuning parameters, or enhancing dataset quality could help achieve more reliable and consistent accuracy across all cases.

8.2. Performance Comparison with Existing Techniques

This thesis was devoted to the problem to distinguish between the cause and effect using only observational data. Numerous studies have been conducted in the past to address this challenge successfully across a variety of datasets. By modeling relationships and optimizing scores like log-likelihood, independence criteria, or divergence measures, different bivariate causal inference approaches which we discussed in Section 6 provide robust frameworks for finding causal structures in all 13 dataset scenarios.

In Figure 8.1, we compare the results of CLS-MML on the real Tübingen cause-effect pairs dataset with other related methods. CLS-MML demonstrates competitive performance, achieving 70% accuracy, showcasing its robustness in handling heteroscedastic noise. While ROCHE slightly surpasses CLS-MML with 71% accuracy , CLS-MML maintains a well-balanced performance across metrics. Compared to LOCIm (62%) and LOCIn (66%), CLS-MML shows a significant improvement, highlighting its capability in causal inference. Notably, our method outperforms HECI (70%), CAM (58%), RESIT (57%), and IGCI (58%), due to its ability to effectively model heteroscedastic noise, which is a common characteristic in real-world dataset.

In Figure 8.2 we show the results for Forced-Decision accuracy metrics of our method compared with other methods for synthetic data sets: AN, AN-s, LS, LS-s and MN-U. Unlike previous studies that reported high accuracy ranging from 80% to 100%, our results did not match these findings. Specifically, the CLS-MML method achieved a performance of only 43% 61% on AN, AN-s, LS and LS-s, which is considerably lower than the 100% accuracy observed in other methods such as ROCHE or LOCI. These results imply that although CLS-MML exhibits promise results, more development is needed

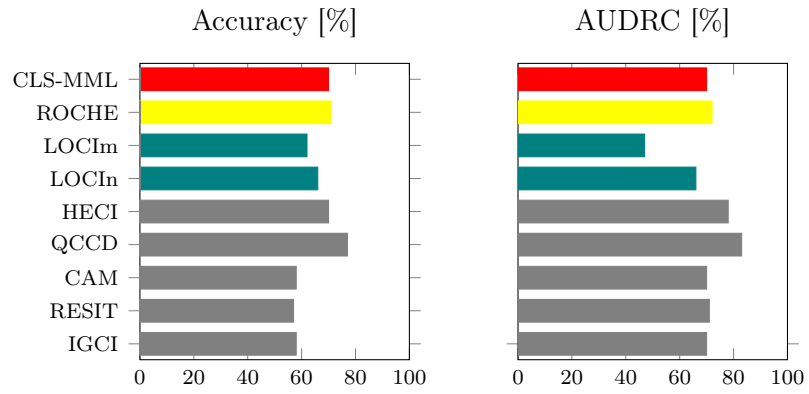

Figure 8.1.: Performance of all methods on Tübingen cause-effect pairs datasets. The best results achieved for the metrics Accuracy, AUDRC are presented for all methods. The methods having heteroscedastic and with homoscedastic noise model.

to match the performance of well-established existing causal inference methods in this field. The Student's t-distribution regression model in the CLS-MML might not capture the underlying data structure if the datasets deviate significantly from its assumptions, such as being heavily skewed or multi-modal. Additionally, the complexity of calculating causality in both directions might not be well-suited for these datasets, especially if the relationships are confounded.

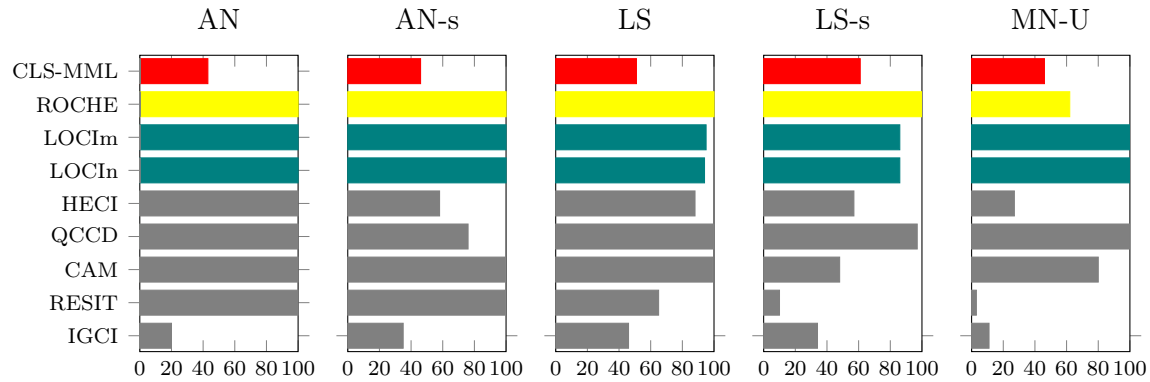

Figure 8.2.: Performance on synthetic cause-effect pairs dataset: AN, ANs, LS, LS-s, MN-u. The best results achieved for the metrics Accuracy, is presented for the methods incorporating heteroscedastic noise models and methods not having heteroscedasticity.

Similarly, the CLS-MML method's performance on the other synthetic dataset like SIM, SIM-c, SIM-ln, and SIM-G, it does not match the high accuracy levels achieved by models like ROCHE and LOCI (see figure 8.3). The metrics values varies between 46%-61%, which is lower compared to other method that achieve a metrics up to 80%. Our method may not be well-suited for these synthetic datasets due to multiple reasons

such as, limitations in modeling complex, non-linear noise structures, handling hidden confounder, and dealing with the complexity of large datasets.

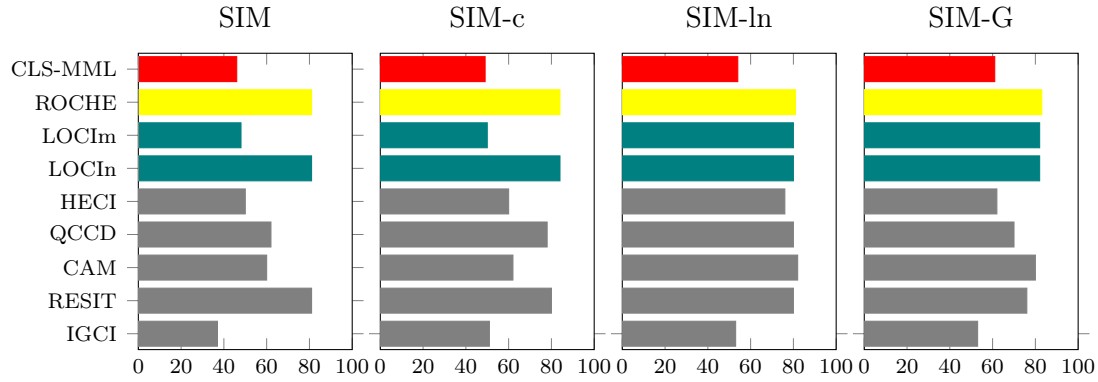

Figure 8.3.: Performance on synthetic cause-effect pairs dataset: SIM, SIM-c, SIM-ln and SIM-G. The average results achieved for the metrics Accuracy and AUDRC are presented for all the methods incorporating heteroscedastic and homoscedastic.

On the other side, CLS-MML outperforms in terms of accuracy and AUDRC on some synthetic dataset. Our proposed approach surpass methods with homoscedastic noise models such as CAM and RESIT on Multi and LSs dataset. For methods with heteroscedastic noise like LOCI, it outperformed for Multi, SIM, Net, including neural networks alternatives. For Net, CLS-MML outperformed only method RESIT. For Cha data set CLS-MML outperformed 6 methods out of 10, including 2 methods using neural networks.

However, a key advantage of CLS-MML is its speed. While other methods like ROCHE or LOCI may take several hours to run (sometimes over 6 hours), CLS-MML converges within 10-20 minutes per dataset. This significant reduction in runtime allows for faster experimentation, making it more efficient and scalable, especially when working with large datasets or when performing multiple iterations for model refinement.

8.3. Limitations and Future Work

Despite its promising performance on certain datasets, CLS-MML has several limitations that should be addressed in future research.

Synthetic data sets like AN-s and MN-U use invertible sigmoid-type functions for the generative process, which may introduce non-linearity and thus it complicates causal direction identification of our algorithm. In the case of SIM-c, which includes a confounder Z, the data generation process introduces additional complexity through the inclusion of an unobserved confounder. Our method may have difficulty handling such latent confounder effectively because it relies on a criterion based on Kolmogorov complexity, which may not be robust enough to distinguish between direct and confounded relationships. The datasets SIM-ln and SIM-G, involve complex noise structures that might not align well with the assumptions of the CLS-MML method. Specifically, SIM-ln incorporates low levels of noise, whereas SIM-G approximates Gaussian distributions for the cause X and employs non-linear additive noise models. The CLS-MML method, which assumes X to be modeled using Student's t-distribution, may struggle to accurately model such low-variance or Gaussian noise, leading to inaccurate causal inferences.

On the other hand, ROCHE and LOCI models, which utilize neural networks, appear to perform best on AN, ANs, LS and LSs synthetic data sets, with nearly perfect results. These techniques may be making use of complex characteristics or underlying structures in the data that perform well in situations where the our method does not. The neural network-based methods may be advantageous due to their ability in learning complex patterns, which could explain their higher precision, even though the precise mechanisms underlying these models are not entirely known.

One limitation of the proposed MML estimation from [WMS18] is that it is designed for small sample size data sets and it struggles with larger datasets. For example, on the Tübingen data pairs, which had fewer than 400 data points, our algorithm was able to identify the correct causal direction almost always. However, as the dataset size increased beyond this threshold, the performance gets weaker. This trend was also observed in most of the synthetic datasets, which had a data value length of 1000 or bigger. The larger sample sizes may introduce more complexity, making it harder to fit Student's t distribution and consequently for the algorithm to maintain its accuracy in identifying causal relationships. This suggests that future improvements could focus on enhancing the algorithm's scalability to handle larger datasets effectively.

Concerning the precision of CLS-MML, the estimation of the robust regression model relies heavily on the parameters ν, τ and μ. The initial fitting of these parameters plays a crucial role, as it directly influences the EM algorithm's ability to find minima of the MML criterion for both causal directions. If the initial estimates are suboptimal, the algorithm may converge to inaccurate solutions, affecting its causal direction identification. This represents a limitation of our method, and future work could focus on developing more robust techniques for parameter initialization to improve accuracy and adaptability across diverse datasets. Furthermore, knowing that real-world data often exhibit deviations from Gaussian distribution, it would be interesting to investigate SCM for other non-Gaussian

distributions.

9. Conclusion

This thesis investigated experimentally the capability of CLS-MML algorithm to detect causal direction between two variables in different domains. Its main focus is to evaluate the effectiveness of using a Student-t robust regression model combined with MML approximation to distinguish the cause and effect between variables. The results demonstrate the practical applicability of the algorithm across various datasets, particularly getting a high accuracy when tested on real-world data.

Contributions This thesis has made contributions both in theoretical advancements and practical implementations of CLS-MML algorithm. The CLS-MML algorithm accounts for heteroscedastic noise in data and uses insights from robust statistics like the Student's t-distribution with a likelihood-based inference score, for accurate parameter estimation. While many related methods perform well on synthetic datasets, they often struggle with the Tübingen pairs. In contrast, our method, despite its simplicity, achieved comparable performance with an accuracy of 70%. The method uses just one hyperparameter — the degrees of freedom — making it simpler to tune and efficient without losing accuracy.

Limitations Although CLS-MML consistently outperformed random baselines in experiments and demonstrated the algorithm's robustness, some limitations remain, particularly in handling data sets with high noise levels. Key insights include the critical role of SNR thresholds and parameter initialization in improving causal direction identification. A limitation of using SNR is that determining the threshold between high and low SNR often requires expert knowledge, making it difficult to use widely. Additionally, while proper parameter initialization is crucial, incorrect initialization can cause the algorithm to converge to local minima, leading to suboptimal results. Having a simple model, though beneficial in some cases, may miss important details in more complex causal relationships, which impacts performance on different data sets.

This work presents a clear, practical approach to handling noisy datasets and provides a foundation for future advancements in bivariate causal modeling. While having the mentioned limitations, the insights gained from this research offer a solid basis for further exploration on potential real-world applications in various domains.

Bibliography

[Tur37] Alan M. Turing. 'On Computable Numbers, with an Application to the Entscheidungsproblem'. In: *Proceedings of the London Mathematical Society* 2.42 (1937), pp. 230–265. DOI: 10.1112/plms/s2-42.1.230. URL: https://doi.org/10.1112/plms/s2-42.1.230.

[NM65] J. A. Nelder and R. Mead. 'A simplex method for function minimization'. In: *The Computer Journal* 7.4 (1965), pp. 308–313. DOI: 10.1093/comjnl/7.4.308.

[Kol68] Andrei Nikolaevic Kolmogorov. 'Three approaches to the quantitative definition of information'. In: *International journal of computer mathematics* 2.1-4 (1968), pp. 157–168.

[WB68] C. S. Wallace and D. M. Boulton. 'An Information Measure for Classification'. In: *The Computer Journal* 11.2 (1968), pp. 185–194.

[Sun74] Rolf Sundberg. 'Maximum likelihood theory for incomplete data from an exponential family'. In: *Scandinavian Journal of Statistics* (1974), pp. 49–58.

[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. 1–38. DOI: 10.1111/j.2517-6161.1977.tb01600.x.

[Ris78] Jorma Rissanen. 'Modeling by shortest data description'. In: *Automatica* 14.5 (1978), pp. 465–471.

[WF87] Chris S. Wallace and P.R. Freeman. 'Estimation and inference by compact coding'. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 49.3 (1987), pp. 240–252.

[LV93] Ming Li and Paul Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer, 1993.

[Byr+95] Richard H. Byrd et al. 'A limited memory algorithm for bound constrained optimization'. In: *SIAM Journal on Scientific and Statistical Computing* 16.5 (1995), pp. 1190–1208. DOI: 10.1137/S1064827591192614.

[CGT00] A. R. Conn, N. I. M. Gould and P. L. Toint. *Trust Region Methods*. MOS-SIAM Series on Optimization. Philadelphia: Society for Industrial and Applied Mathematics, 2000. DOI: 10.1137/1.9780898719857.

[Pea00] Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2000.

[Gre+05] Arthur Gretton et al. 'Measuring statistical dependence with Hilbert-Schmidt norms'. In: *International conference on algorithmic learning theory*. 2005, pp. 63–77.

[Wal05] Chris S. Wallace. *Statistical and Inductive Inference by Minimum Message Length*. Springer, 2005.

[RW06] Carl E. Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. Vol. 1. Cambridge: MIT Press, 2006.

[Shi+06] S. Shimizu et al. 'A linear non-Gaussian acyclic model for causal discovery'. In: *Journal of Machine Learning Research* 7 (2006).

[Hoy+08] Patrik Hoyer et al. 'Nonlinear causal discovery with additive noise models'. In: *Advances in neural information processing systems* 21 (2008).

[JS10] Dominik Janzing and Bernhard Schölkopf. 'Causal inference using the algorithmic Markov condition'. In: *IEEE Transactions on Information Theory* 56.10 (2010), pp. 5168–5194.

[Ste+10] Oliver Stegle et al. 'Probabilistic latent variable models for distinguishing between cause and effect'. In: *Advances in Neural Information Processing Systems* 23 (2010).

[Jan+12] Dominik Janzing et al. 'Information-geometric approach to inferring causal directions'. In: *Artificial Intelligence* 182 (2012), pp. 1–31.

[ZH12] Kun Zhang and Aapo Hyvarinen. 'On the identifiability of the post-nonlinear causal model'. In: *arXiv preprint arXiv:1205.2599* (2012).

[BL13] K. Bache and M. Lichman. *UCI Machine Learning Repository*. `http://archive.ics.uci.edu/ml`. 2013.

[BPE14] Peter Bühlmann, Jonas Peters and Jan Ernest. 'CAM: Causal additive models, high-dimensional order search and penalized regression'. In: (2014).

[Pet+14] Jonas Peters et al. 'Causal discovery with continuous additive noise models'. In: *JMLR* (2014).

[Sgo+15] Eleni Sgouritsa et al. 'Inference of cause and effect with unsupervised inverse regression'. In: *Artificial intelligence and statistics*. PMLR. 2015, pp. 847–855.

[Moo+16] Joris M Mooij et al. 'Distinguishing cause from effect using observational data: methods and benchmarks'. In: *Journal of Machine Learning Research* 17.32 (2016), pp. 1–102.

[MV17] Alexander Marx and Jilles Vreeken. 'Telling cause from effect using MDL-based local and global regression'. In: *2017 IEEE international conference on data mining (ICDM)*. IEEE. 2017, pp. 307–316.

[PJS17] Jonas Peters, Dominik Janzing and Bernhard Schölkopf. *Elements of Causal Inference: Foundations and Learning Algorithms*. Cambridge, MA: The MIT Press, 2017. ISBN: 9780262037310. URL: `https://mitpress.mit.edu/9780262037310`.

[Gou+18] Olivier Goudet et al. 'Learning functional causal models with generative neural networks'. In: *Explainable and Interpretable Models in Computer Vision and Machine Learning* (2018), pp. 39–80.

[WMS18] Chi Kuen Wong, Enes Makalic and Daniel F Schmidt. 'A minimum message length criterion for robust linear regression'. In: *arXiv preprint arXiv:1802.03141* (2018).

[GSB19] Isabelle Guyon, Alexander Statnikov and Berna Bakir Batu. *Cause effect pairs in machine learning*. Springer, 2019.

[TCV20] Natasa Tagasovska, Valérie Chavez-Demoulin and Thibault Vatter. 'Distinguishing cause from effect using quantiles: Bivariate quantile causal discovery'. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 9311–9323.

[DN22] Bao Duong and Thin Nguyen. 'Bivariate causal discovery via conditional divergence'. In: *Conference on Causal Learning and Reasoning*. PMLR. 2022, pp. 236–252.

[Imm+22] Alexander Immer et al. 'On the Identifiability and Estimation of Causal Location-Scale Noise Models'. In: *arXiv preprint arXiv:2210.09054* (2022).

[Xu+22] Sascha Xu et al. 'Inferring cause and effect in the presence of heteroscedastic noise'. In: *International Conference on Machine Learning*. PMLR. 2022, pp. 24615–24630.

[Imm+23] Alexander Immer et al. 'On the identifiability and estimation of causal location-scale noise models'. In: *International Conference on Machine Learning*. PMLR. 2023, pp. 14316–14332.

[HM24] Katerina Hlaváčková-Schindler and Suzana Marsela. *Causal Location-Scale Noise Models by Minimum Message Length*. Technical Report. Faculty of Computer Science, University of Vienna, 2024.

[Tra+24] Quang-Duy Tran et al. 'Robust Estimation of Causal Heteroscedastic Noise Models'. In: *Proceedings of the 2024 SIAM International Conference on Data Mining (SDM)*. SIAM. 2024, pp. 788–796.

[Wik24a] Wikipedia contributors. *Signal-to-Noise Ratio*. `https://en.wikipedia.org/wiki/Signal-to-noise_ratio`. Accessed: 2024-12-06. 2024.

[Wik24b] Wikipedia contributors. *Signal-to-noise ratio*. Accessed: 2024-01-04. 2024. URL: `https://en.wikipedia.org/wiki/Signal-to-noise_ratio`.

Acronyms

AMI Algorithmic Mutual Information. 9

ANM Additive Noise Model. 7

AUDRC Area Under the Decision Rate Curve. 2, 33, 54

BMI Body Mass Index. 21, 49

CLS-MML Causal Location Scale Minimum Message Length. iii, v, xi, xiii, 2, 25, 32, 34, 35, 43, 46, 49–55, 57

DAGs Directed Acyclic Graphs. 6, 39

EM Expectation-Maximization. 14, 25, 26, 31, 32, 35–37, 46, 55

GNI Gross National Income. 47

GP Gaussian Processes. 17, 19

HNMs Heteroscedastic Noise Models. 7

HSIC Hilbert-Schmidt Independence Criterion. 40

KC Kolmogorov Complexity. iii, v, 9–11, 13

LSNM Location Scale Noise Model. iii, v, 2, 7, 8, 40

MDL Minimum Description Length. vii, 1, 10, 11, 40

MLE Maximum Likelihood Estimates. 26, 27, 30, 37, 47, 48, 51

MML Minimum Message Length. iii, v, vii, 1, 10, 11, 13–16, 23, 33, 49, 55, 57

RMSE Root Mean Square Error. 15

SCM Structural Causal Model. iii, v, 2, 5–7, 55

SEM Structural Equation Modeling. 5

SNR Signal-to-Noise Ratio. iii, v, viii, xi, xiii, 2, 16, 25, 26, 35–37, 43–47, 51, 52, 57

A. Appendix