University of Tunis

Higher Institute of Management of Tunis

Master Thesis in IT applied to the management

Integrating Ontological Knowledge for Iterative Causal Discovery and Visualization

By:

Montassar Ben Messaoud

Defence on: May 21, 2009

Supervisors:
Pr. Philippe **LERAY** (Polytechnic Institute of Nantes University-France)
Dr. Nahla **Ben Amor** (Higher Institute of Management of Tunis-Tunisia)

To my loving parents,
to my dear family members,
to my faithful friends.

Acknowledgements

First of all, I would like to express my gratitude to all members of the jury who agree to judge my dissertation.

I am grateful to my supervisor Dr. Nahla Ben Amor for her patience and encouragement throughout my graduate studies. Her technical and editorial support was essential to the completion of this dissertation and has taught me innumerable lessons and insights on the workings of academic research in general.

I have been very fortunate for having Pr. Philippe Leray as my research advisor. Many thanks to him for introducing me to an interesting research area and inviting me to work in France, where my thesis was partially achieved. My association with him has helped me grow not only as a researcher but also as an individual.

I am indepted to my professor Boutheina Ben Yaghlane who kindly accepted to have interesting discussions with me.

Last but not least, I want to thank from the bottom of my heart my parents for their understanding, support and encouragement in my most difficult times. Their infinite love and prayers have been my greatest strength all these years.

Special thanks to my closest friends who continuously encouraged, conforted and motivated me.

Abbreviations & Acronyms

BIC	Bayesian Information Criterion
BN	Bayesian Network
CBN	Causal Bayesian Network
CPD	Conditional Probability Distribution
CPDAG	Complete Partially Directed Acyclic Graph
DAG	Directed Acyclic Graph
D_O	Observational dataset
D_E	Experimental dataset
XML	Extensible Markup Language
FCI	Fast Causal Inference
GES	Greedy Equivalence Search
GO	Gene Ontology
GS	Greedy Search
IC	Inductive Causation
I-map	Independence Map
JT	Junction Tree
JPD	Joint Probability Distribution
MCMC	Monte Carlo Markov Chain
MDL	Minimum Description Length
ML	Maximal Likelihood
MWST	Maximum Weight Spanning Tree
MyCaDo	My Causal DiscOvery
OWL	Web Ontology Language
PDAG	Partially Directed Acyclic Graph
RDF	Resource Description Framework
SemCaDo	SEMantical CAusal DiscOvery
SEM	Structural Equation Models
SGS	Spirtes, Glymour & Scheines
PC	Predictive Causation
UnCaDo	Unsure Causal DiscOvery

Contents

Introduction **1**

1 Foundations of Causal Theory **3**

1.1 Introduction 3

1.2 Causality from the beginning 4

1.3 Basic principles of causal theory 7

1.3.1 Causation vs Association 7

1.3.2 Real cause vs spurious cause 10

1.3.3 Mechanisms 11

1.3.4 Counterfactuals 13

1.3.5 The science of seeing Vs The art of doing: 14

1.3.6 Observational vs interventional data 16

1.4 Causal modeling framework 17

1.5 Conclusion 18

2 Causal Bayesian Networks **19**

2.1 Introduction 19

2.2 Notations and definitions 20

2.2.1 Notations 20

2.2.2 Definitions 20

2.3 Bayesian networks 23

2.3.1 Definition 23

2.3.2 The d-Separation criterion 25

2.3.3 Inference with Bayesian networks 27
2.3.4 Learning Bayesian networks 28
2.4 Causal Bayesian networks 35
2.4.1 Definitions and properties 35
2.4.2 The "do" operator 36
2.4.3 Conditioning Vs manipulating 36
2.5 Learning CBNs 39
2.5.1 Active learning for CBN structure 39
2.5.2 Learning CBN from mixture of observational and experimental data 40
2.5.3 Decision theoretic approach for learning CBNs 40
2.6 Conclusion 50

3 New Strategy for Iterative Causal Discovery & Visualization 51
3.1 Introduction 51
3.2 Basics on Ontologies 52
3.3 Integrating ontological knowledge for causal discovery 56
3.4 Integrating ontological knowledge for causal graph visualization 59
3.5 Toy Example 63
3.6 Conclusion 67

4 Implementation and Experimentations 68
4.1 Introduction 68
4.2 Programming Tools 69
4.3 Implementation 69
4.3.1 Ontology & Random CPDAG Generation 70
4.3.2 Interventional Data Sets Generation 73
4.4 Complexity Analysis (MyCaDo Vs SemCaDo) 76
4.5 Experimental results 77
4.6 Conclusion 78

Conclusion 79

Bibliography **81**

List of Figures

1.1 Causation Vs Association [47] 9
1.2 Confounding causal relationship 10
1.3 Counterfactual Vs Actual World [47] 14

2.1 a v-structure 22
2.2 (a) a singly connected DAG, (b) a multiply connected DAG 22
2.3 (a) example of DAG, (b) the skeleton relative to the DAG and (c) an example of a PDAG. 23
2.4 Example of DAG (V = X1, X2, X3) 24
2.5 The three basic connection structures 26
2.6 Example of DAG (V = X_1, X_2, X_3, X_4, X_5) 26
2.7 When there is a correlation between X_i and X_j, one of these three causal structures hold. 29
2.8 Markov equivalence 30
2.9 ASIA theoretical network 31
2.10 Run example of PC 31
2.11 Main description of the MyCaDo approach [35] 41
2.12 Toy CBN (left) and the corresponding CPDAG (right) 46
2.13 All possible instantiations for $Xi - Ne_U(Xi)$, the possible structures compatible with this instantiation and the result of inferring edges. 49

3.1 Example of hierarchy representing a taxonomy of concepts. 54
3.2 All possible instantiations for $X2 - Ne_U(X2)$ 58

3.3 (a) An example of graph; (b) An example of applying a Force Placement Algorithm to the initial graph. 60
3.4 an example of tree hierarchy 62
3.5 Two graph visualizations correponding to level 3 (a) and level 1 (b) of the tree hierarchy presented in Figure 3.4 62
3.6 Main description of the SemCaDo approach 63
3.7 An example of CPDAG 63
3.8 All possible instantiations for $Xi - Ne_U(Xi)$, the possible structures compatible with this instantiation and the result of inferring edges. 65
3.9 an example of graph correponding to level 3 (a) and level 2 (b) of the tree hierarchy presented in Figure 3.4 66

4.1 CBN, CPDAG and Obseravational Data generation 70
4.2 An example of protégé output 71
4.3 An example of graph mutilation 74
4.4 Experimental data generation from a known CBN 75
4.5 The number of experiments needed given the number of nodes in the graph 77

List of Tables

1.1 Sketch of some important mechanisms in science [51]. 12

2.1 A priori probabilities 24
2.2 Conditional probabilities 25

3.1 Rada et al.'s distance matrix between the nodes given in Figure.3.1 56

Introduction

Over the last 20 years, Bayesian Networks [23] have become a popular representation for encoding uncertain knowledge in expert systems. The BNs have been used for prediction and classification in various domains such as medical diagnostics, financial data analysis, pattern recognition, fraud detection and bioinformatics.

In the first application, the BN structure was causally defined by expert knowledge. Then, algorithms were proposed in order to learn the BN structure from observational data. Generally, these algorithms can only find a structure encoding the right conditional independencies but not all the causal relationships.

Some new domais appear where the model will only be learnt in order to discover these causal relationships. To this end, we will focus on discovering causal relations in order to get Causal Bayesian Networks (CBN).

In fact, CBNs reflect more intuitive, comprehensible and informative structure than standard BNs, as every edge connecting variables represents an autonomous cause-effect relationship. This is not usually true with standard BNs since they represent symmetric dependences between variables instead of causal ones. Thus, CBNs can be considered as an extension of standard BNs allowing causal reasoning [41] (i.e. predicting the effect of external interventions using causal inference) in addition to evidential one [26, 40].

To learn such models, observational data are insufficient to discover the true causal

structure, therefore interventional data (i.e. samples conditioned on the particular values of one or more variables that have been experimentally manipulated) are required.

These interventions performed on variables are usually very expensive to perform, time-consuming or even impossible to perform. Hence the choice of variables to experiment on can be vital when the number of experimentations is restricted. In other words, the experiments should be selected with care and following a well-defined strategy taking into account the available resources and the informative contribution of such interventions.

On the other hand, available ontologies provide high level knowledge for the same domain under study. Consequently, using the semantical knowledge contained in the ontologies can turn out of a big utility to improve causal discovery.

In this work, we propose a new method for learning CBNs from observational data and interventions. We first extend the greedy approach proposed in [35], by adding a new step based on the integration of ontological knowledge, which will allow us to choose efficiently the interventions to perform in order to obtain the complete CBN. Then, we propose an enriched visualization for better understanding of large causal networks containing a huge number of variables.

This report is organized in four chapters: As everyone has his intuitive feel of what a causal relationship is, we will start the first chapter by explaining the basic ideas of causal theory. In Chapter 2 we will discuss our modeling requirements and explores the state-of-the-art on learning CBNs. Next, we present our resarch proposition and an overview of the contributions. Chapter 4 deals with the implementation and the evaluation of our approach.

Chapter 1

Foundations of Causal Theory

1.1 Introduction

Over the last few years, the researches in the artificial inteligence field focus on the creation of autonomous inteligent systems able to build workable models, reason over them and make sense of what goes on in their environment, so they can decide to react for different situational tasks.

The notion of causality (based on experimentation) was introduced since more than 200 years and has long been recognized as an important research axis. At the same time, several methodologies have been proposed and a well-defined framework was developed, allowing the representation of phenomena based on the introduction of the notion of interventions.

In this chapter, we first review the historical steps which marked the evolution of causality and then introduce basic concepts of causation which unites the axis of mathematics to philosophy.

1.2 Causality from the beginning

Causality is relative to the relationship between cause and effect, i.e. the way we understand the world. Metaphysically, **Aristotle** (384 BC-322 BC) described four levels of causality which can be applied to the relationships between phenomena and their causes:

1) *Physical cause* - All the things from which an effect is produced including parts, constituents, substratus, materials.

2) *Formal cause* - The examination of cause in terms of teleology and the function of effects in the context of a greater whole (part-whole causation).

3) *Efficient cause* - The description of an effect's cause from the standpoint of external agency or causal events. An efficient cause can be limited to apparent or relevant causal events.

4) *Final cause* - An explanation of effect in terms of ultimate cause(s), from the standpoint of the purpose, or end, that a given effect is observed to serve. Final causation is taken a step further than Formal causation in terms of an overall system, and usually refers to a philosophical or metaphysical framework as opposed to a practical or utilitarian framework.

Bacon in 1620 shared the same idea about the real existence of causality and believed that the Aristotle *four causes* should be partitioned into two sets, physics that gathers material and efficient causes, and metaphysics that deals with the formal and the final causes.

But more precisely the modern study of causation begins later with the Scottish philosopher **David Hume** (1711-1776). Hume has introduced to philosophy three revolutionary ideas that, today, are taken for granted by almost everybody, not only philosophers.

On the Introduction of *A Treatise of Human Nature*, Hume writes that "the science of man is the only solid foundation for the other sciences", and that the correct method for

this science is "experience and observation" i.e. the empirical method. Also, he made a sharp distinction between analytical and empirical claims. He classified causal claims as empirical, rather than analytical and, in the same time, identified the source of all empirical claims with human experience.

According to Hume, the notion of causation is closely linked to the problem of induction. Generally, we reason inductively by associating conjoined events, and it is the mental act of association that is the basis of our concept of causation.

We can, for instance, mention the famous Hume's three conditions for inferring cause:

1) ***Contiguity***, nearness or contact between presumed cause and effect;

2) ***Temporal precedence***, here the cause had to precede the effect in time;

3) ***Constant conjunction***, in that the cause had to be present whenever the effect was obtained.

In addition, Hume argues that induction is founded on the persistance of regularities (sometimes called the Uniformity of Nature) and that we cannot affirm that nature is uniform by applying these two types of reasoning:

1) Demonstrative reasoning (effectively, deductive reasonning)

2) Probable reasonning (effectively, inductive reasonning)

With regards to (1), Hume argues that we cannot prove a priori that regularities will continue, as it is " consistent and conceivable" that the course of nature might change. Coming to (2), Hume argues that founding a regularity on the fact that regularity has always operated in the past (inductive reasonning) is arguing in a circle, because induction was the very process we were trying to explain in the first place. Hence no form of reason will sponsor inductive inference. That's why a great debate on the problem of induction emerged at this time.

In his famous book "*A Treatise of Human Nature*", Hume used an example which illustrates perfectly the reasoning with cause and effect. He asserted that private property is not a natural right, but he argues that it is justified since resources are limited. Consequently, if all goods were unlimited and available freely, then private propoerty would not be justified.

Hume also influenced many philosophers as **Emmanuel Kant**(1724-1804) with awakening him from his "dogmatic slumber", since he radically changed the foundations of methaphysics by introducing the concept of experimentation.

An other influent contribution in causal inference, was the work of **John Stuart Mill** in 1843, who demonstrated that we cannot prove the existence of causality without experimentation and he proposed four methods for establishing causality, namely, the concomitant method, the method of difference, the method of residues and the method of agreement.

In the other hand **Bertrand Russel**(1872-1970) contended that if we go with Hume in rejecting the rationality of induction from empirical data then scientists are forced to reject all empirical method or concede the irrationality of science.

Karl Popper(1902-1994) follows an other road by showing how science could be rational by not using induction and consistent with empiricism in the same time. According to Popper, the only kind of reasoning in science is deductive. Thus, he will support the falsification's criterion in spite of verification as a solution to the problem of induction. Using the famous example, the observation of a single black swan, disproves the statement that "*all swans are white*". Hence, falsifiability (or "refutability") was interpreted as the possibility that an assertion can be shown false by an observation or a physical experiment.

We will wait until the mid 1980s, when Judea Pearl, Dan Geiger and Thomas Verma, computer scientists at UCLA working on the problem of storing and processing uncertain information in intelligent agents, who realized that uncertain information could be stored much more efficiently by taking advantages of conditional independence, and by using di-

rected acyclic graphs to encode probabilities and the conditional independence relations among them.

In the late 1980s, Peter Spirtes, Clark Glaymour and Richard Scheines, a team working on the problem of causal inference at the Philosophy Department in the Carnegie Mellon University, combined the artificial intelligence work of Pearl and his colleagues to the problem of testing and discovering causal structure in behavioural sciences. And some conditions and assumptions under which reliable inference about the effects of manipulations is possible have been precisely characterized. Thus the causal bayesian networks become a reality and a popular tool for inferring and discovering causal structures.

1.3 Basic principles of causal theory

Reasoning with cause and effect seems to be an easy theory for much people. It is due to surely having common sense ideas about causation. To this end, this section was included since working through this theory, need to approach learning the definition of every notion in causation like you would approach learning a definition in physics or mathematics.

For example, most people have the intuitive idea of "momentum", but physicists use the term in a very precise way. The same is true about the idea of "causation", every term should be used in his precise context and avoid any confusion.

1.3.1 Causation vs Association

One of the hard tasks in causation is to understand and fix the borders between observed associations and causal associations. Causation and association are indeed quite different ideas, and the subject of causal and statistical reasoning depends on carefully separating them. They are, however, related.

First, statistical association by itself does not prove causation[47], however, causal relation can produce, automatically, statistical correlation. To illustrate this idea, let us show

an example. Right eye and left eye color are highly associated, but neither is the cause of the other. To say they are highly associated means that if you were to observe that someone's left eye was blue, you would be in good position to say that their right eye was also of the same color. If we were to intervene to surgically replace a random person's left eye with a green one, however, their right eye would not change color to green. That is the first difference between causation and association.

Otherwise, we should remember that the theory of causation is based, essentially, on establishing stable mechanisms or physical laws. The following example stresses on the need to discover the hidden mechanisms to take into consideration the statistical correlations as causal.

Example 1 *Lung cancer rates increased significantly in Great Britain and the United States during the first half of the twentieth century, correlating with increase in smoking, but carefully controlled studies only began to appear in the 1950s [25]. In one classic study conducted in England, 649 male and 60 female patients with lung cancer were matched to an equal number of control patients of the same age and sex. For both men and women, there was a strong correlation between lung cancer and smoking, particularly heavy smoking. By 1964, when the U.S. Surgeon Generals Report asserted a causal link between lung cancer and smoking, there had been 29 controlled studies performed in numerous countries that showed a high statistical association between lung cancer and smoking. Although the exact mechanism by which smoking causes cancer was not known. That's why, up till now, the U.S. government can't take any corrective measurements against the tobacco's industrials. According to this example, correlations can provide evidence for causes but does not prove causation.*

Figure 1.1: Causation Vs Association [47]

The causal association between an effect and a possible cause can be measured by:

$$\Delta P_c = P(\text{Effect} \mid \text{Exposure to Cause}) - P(\text{Effect} \mid \text{Removing Cause}) \tag{1.1}$$

The problem in this step or rather the major scientific challenge is that P(Effect | Removing Cause) is the probability relative to the Counterfactual population which is unobservable, because we can not go back in time and remove the cause from the actual exposed population.

However we dispose of an observed association which is measured by:

$$\Delta P_c = P(\text{Effect} \mid \text{Exposure to Cause}) - P(\text{Effect} \mid \text{No Exposure to Cause}) \tag{1.2}$$

The statistical association is usually used in identifying the causes of diseases but mechanism considerations are also often relevant to assessing causality.

Also, we should remember one of the more fundamental properties of association, the *symmetry*. Association is a symmetric relation. If one property, like wealth, is associated with another, like education, then education is also associated with wealth. In other words, if learning that an individual is wealthy changes your certainty about his or her level of education, then learning about an individual's educational level first is also informative about his or her wealth. This is true generally for any properties P1 and P2. If P1 and P2 are associated, then P1 is informative about P2, and P2 is informative about P1.

However causation is *asymmetric*. For example, whatever causation is, it has a direction. Warm weather causes people to wear lighter clothing, but wearing lighter clothing does not cause warm weather. A theory that fails to capture the asymmetry of causation will be unsatisfying[46].

We should clarify that saying that causation is asymmetric does not mean that symmetric cases of causation are prohibited. It only means they are not necessary. In some cases causation can go both ways. For example, losing sleep can cause anxiety, and anxiety can also cause a loss of sleep.

Figure 1.2: Confounding causal relationship

1.3.2 Real cause vs spurious cause

One philosophical explanation about causality consists in finding the real meaning of the sentence A actually causes B. Inspired by the ideas of I. Good [20] about determining responsabilities, Pearl and Halpern [22] tried to distinguish two sorts of causes, the real cause (i.e actual cause) and spurious cause.

In order to more illustrate the problem, let us consider two fires advancing toward a house. If fire A burned the house before fire B, we would consider fire A "the actual cause" for the damage, even supposing the house would have definitely burned down by fire B.

Classifying the two fires equally as a contributory cause is counterintuitive. Here the second cause becomes ineffective only because the effect "has already happened". Thus a temporal notion should be taken into consideration in the developed models to generate the correct results. Such models will be called *Dynamic Causal Models*.

An other problem which faced the causal theory and that goes beyond mere philosophical speculation was the confounding factor, which raises on the presence of a third event that causes the two others, as illustrated by Figure 1.2. Considering one thing that can occur before an other as cause and effect is a trap, because they are effects of a common cause, a phenomenon known as *spurious causation*. For example, flashes of lighting appear just before and seem to cause the thunderclaps that follows them, but in reality both are effects of a coomon cause: the superheating of air molecules from the massive static electric discharge between the earth and the atmosphere.

Note that a good theory of causation ought to successfully separate cases of real from spurious causation.

1.3.3 Mechanisms

The world is organized in the form of stable mechanisms, or physical laws, which are sufficient for determining all events that are of interest to the modeler. Properly speaking, a mechanism is a system of parts that operate or interact like those of a machine, transmitting forces, motion, and energy to one another. The mechanisms are autonomous, like mechanical linkages in a machine, or logic gates in electronic circuits. As a result, we can change one without changing the others.

In the sixteenth and seventeenth centuries, natural philosophers try more and more to understand the world in terms of mechanisms, since the born of the Newton's unified explanation of the gravitational interaction of objects (See Table 1.1).

In the history of science, progress has been made in many domains by the discovery of new mechanisms. That is why discovering mechanisms is one of the basic principles of the causality.

Example 2 *Consider the case of Julia, a 50 years old woman who goes to her doctor complaining of stomach pains. After ordering some tests, the doctor tells here that she has a gastric ulcer. If this were the 1950s, the doctor would probably tell her that she needs to take it easy and drink more milk. If this were the 1970s or 1980s, Julia would probably be told that she suffered from excessive acidity and be prescribed Zantac or similar antacid drug. But since 1990s, when the bacterial theory of ulcers were developed, her well-informed doctor tells her that she probably has been infected by a newly discovered bacterium called Helicobacter pylori and that she needs to take a combination of antibiotics that will eradicate the bacteria and cure the ulcer.*

Medical researchers are highly concerned with finding mechanisms that explain the occurrence of diseases. Thus, understanding the mechanism that produces a disease can lead to new ideas about how the disease can be treated.

To understand the role of mechanims in causal models, let us consider a circuit diagram. First of all, it is qualified as a causal model because it contains the information to confirm

Table 1.1: Sketch of some important mechanisms in science [51].

Science	**Parts**	**Changes**	**Mechanisms**
physics	sun & planets	motion	forces, gravitation
chemistry	elements, molecules	mass, energy	reactions
genetics	genes	genetic transmission	heredity, mutation
geology	geological formations	creation & elimination of formations	volcanic eruptions
plate tectonics	continents	motion such as continental drift	collisions
neuroscience	neurons	synaptic connections	electrochemical transmissions
cell biology	cells	growth	cell division

or refute all action. For example, anyone can figure out what the output would be like if we change an OR gate to a NOR gate in the circuit. Here, the symbolic units correspond to stable physical mechanisms (i.e. the logical gates). Consequently, each variable, has precisely only one mechanism that determines its value and connect it to its parents in the model. And the more interesting is that if we know which mechanisms each action modifies, and the nature of the modification, we can avoid all talk of causation.

1.3.4 Counterfactuals

The basic idea of counterfactual theories of causation is that the meaning of a singular causal claim of the form "*Event a caused event b*" can be explained in terms of counterfactual conditionals of the form "*If a had not occurred, b would not have occurred*".

Analyses along these lines have become popular in the last quarter of the twentieth century, especially since the development in the 1970's of possible world semantics for

Figure 1.3: Counterfactual Vs Actual World [47]

counterfactuals. The best known counterfactual analysis of causation is David Lewis's theory.

Lewis' theory rests on two ideas: the existence of alternative *possible worlds*, and a similarity metric over these worlds. For Lewis, the meaning and truth of counterfactuals depend on our similarity metric over possible worlds. When we say "*if A hadn't happened, then B wouldn't have happened either*", we mean that for each possible world W1 in which A didn't happen and B did happen, there is at least one world W2 in which A didn't happen and B didn't happen that is closer to the actual world than W1.

Two questions were posed within the hypothetical example as a means of illustrating the basic idea of this section. Was malignant brain cancer caused by the exposure to the radio frequency signals emitted by the phone? And if the person affected by cancer had not been exposed to his GSM cell phone, he would not have gotten brain tumor. How are we to imagine his life without any radio frequency signals?

We need to conceive a circumstance in which the *only* difference between the actual world and the hypothetical world is the presence or absence of the studied cause. This counterfactual perspective, made famous by Donald Rubin (1974), makes it clear why causal claims are so hard to establish scientifically. Effectively, we can not go back in time and let people re-live their lives without any exposure to radio signals, nor can we go back in time and remove, completely, the use of telecommunications.

In Figure 1.3, Actual Population 1 represents the actual population exposed massively to radio frequency signals in urban area. The counterfactual Population 1 represents a counterfactual world we can not access: a world involving the same persons living the same life, save that we have gone back in time and intervened to remove the all radio signals from their environment. The Actual Population 2 represents what we can obtain

from another group of persons living, for example, in rural areas. The problems, of course, is that such a group will inevitably differ in lots of ways from our first tested sample.

1.3.5 The science of seeing Vs The art of doing:

If we look at the history of science, we find that it was based on two components: One consisting of passive observations (seeing), and the other consisting of voluntary intervention or actively changing something (doing). For example, *a mum might observe her son putting on weight, or she might intervene by starting a new regime for him.*

But algebra was not equally fair to these two components. Mathematical techniques were developed exclusively to support the former (seeing) not the latter (doing) [41].

By combining the two components, the laboratory is the ideal place, where the "Seeing" part enjoys the power of algebra, whereas the "Doing" part depends on scientist's judgement. When an intervention changes a DNA's combination from one state to another, a new set of algebraic equations becomes applicable, but, still now, there is no mathematical operations to represent the transfer from one combination to another. Thus, we need an algebra of doing in addition to the actual seeing algebra.

Let us take a simple example to clarify the difference between the two algebra. If we wish to find the probability that it rained, given that we "see" the grass wet, we can express our question in a formal way by using the probability theory to transform the sentence into other expressions that are more convenient or informative.

Here, by referring to the Bayes rule, we can respond automatically:

$$P(rain/wet) = P(wet/rain) \times \frac{P(rain)}{P(wet)} \tag{1.3}$$

But suppose we ask a different question: "What is the chance it rained if we *make* the grass wet?" The syntax of probability will be unable to express our query, because the vertical bar in the conditional probability is already taken to mean "given that we see".

We know intuitively what the answer should be: P(rain), because making the grass wet does not change the chance of rain. But can this intuitive answer, and others like it, be derived mathematically?

A solution exists, and it takes a new algebra, using the *do(x)* operator, for which we have a new semantics in terms of surgeries. The do operator will be reviewed in details in the next chapter.

1.3.6 Observational vs interventional data

By referring to the Oxford Dictionary we find that the *observation*'s term is defined as "*the act of watching*". In other words, it is a detailed examination of something before analysis, diagnosis, or interpretation.

Scientifically speaking, an observation characterizes evidence for the presence or absence of an organism or set of organisms through a data collection event at a location, as defined by the Taxonomic Data Working Group's (TDWG) Observational Data sub-group. Here we will focus on observational data as the major tool for seeing.

The same Dictionary defines the word "*intervention*" as a scientific procedure undertaken to make a discovery, test a hypothesis, or demonstrate a known fact. Thus an intervention is synonymous of an action tentatively adopted without being sure of the outcome. Generally, valuable experiments is that which can be reproduced by a variety of different investigators and lends to theoretical analysis.

So which contribution did we expect by using this kind of data and what is the difference with regard to the observational data mentioned above?

Example 3 *To answer these questions, let us take the example of the laboratory of fire experiements. The main mission of this laboratory consists of the realization of experimental programs in which the produced data are of use to the development and validation of the means of calculations in the Institute of radioprotection and nuclear power's safety. The*

realized experiments implement the knowledges and the professions relative to the following domains: Physical appearance of fires in stuffy and ventilated circles (thermal of the fire, the distribution of fires, smokes, gases, etc.), Controlling the experiment with average and large scale (buildings, technical premises, etc.) and all the specific instrumentation (temperature, stream of heat, concentration of gases, soots, pressure, debit).

Consequently, performing scientific experiements allows us to make interventions and test the validity of hypothesis. We should note that all experiments must be led in the respect for the scientific ethics and in the respect for the security of the persons and the environment. For example, the national and international laws prohibited to make any vivisection (experimental surgery on a living organism) without having anesthetized the animal.

1.4 Causal modeling framework

We view in the last section how nature can be expressed in the form of stable causal mechanisms. These mechanisms are organized in the form of an acyclic structure, which scientist attempts to identify from available observations using models [41].

A *model*, in the common use of the word, is an idealized representation of reality that highlights some aspects and ignores others. In logical systems, for example, a model is a mathematical object that assigns truth values to sentences in a given language, where each sentence represents some aspects of reality. A causal model, naturally, should encode the truth values of sentences that deal with the causal relationships. Pearl distinguished three major sentences:

- Action sentences: B will be true if we do A.
- Counterfactuals: B would be different if A were true.
- Explanation: B because of A.

Such models are well suited to encode and distinguish information about external changes through an explicit representation of the mechanisms that are altered in such

changes.
In the literature, we can distinguish two principal models to express causality:

- **Functional Causal Models:** In those models, causal relationships are expressed in the form of *functional* and *deterministic* equations, through the assumption that certain variables in the equations are unobserved.

$$\forall i \in \{1, ..., n\}, \qquad x_i = f_i(pa_i, u_i) \tag{1.4}$$

  where pa_i (connoting *parents*) stands for the set of variables judged to be immediate cause of x_i, and u_i represents errors (or disturbances) due to omitted factors.

  A set of equations in the form of (1.4) where each equation represents an autonomous, active mechanism is called structural model. If each mechanism determines the value of just one distinct variable (i.e. dependent variable), then the model is called *Structural Equation Model* (SEM) [41].

- **Causal Bayesian Networks:** which represent an extension of the traditional Bayesian networks since each directed edge represents an autonomous causal relation [41]. We will focus, minutely, on the study of Causal Bayesian Networks in the next chapter.

Both models have the same representational power and the use of the one or the other depends on the nature of the problem.

1.5 Conclusion

In this chapter, we have introduced some basic concepts, notations and definitions related to causal theory. We also tried to make a compromise between the philosophical aspect of causality and mathematics.

Moreover we showed that there exist two main formalisms for causal modeling, the Structural Equation Models and the Causal Bayesian Networks. The use of one model instead of the other depends on the domain. In general, SEMs are suited for confirmatory studies, while CBNs are suitable for exploratory research.

In our work, we will focus on the Causal Bayesian Network by studying them minutely in the next chapter.

Chapter 2

Causal Bayesian Networks

2.1 Introduction

Bayesian networks were introduced in the 1980's as a formalism for representing and reasoning with models of problems involving uncertainty, adopting probability and graph theory as a basic framework [40].

Over the last few years, several researchers have proposed algorithms to learn BN structures from purely observational data. However, it has been proved that only the equivalence class of the underlying structure can be discovered. This implies a randomly edge orientation to fully orient the partially directed structures.

To end with this limitation, an extension of BNs were introduced, where the semantics of edges are viewed as autonomous causal relations. The Causal Bayesian Networks (CBNs) need, however, additional data to fully determine the true causal structure. More precisely, they extract causal knowledge from performing real experiements on the system under study. Several approaches and techniques have been developed to handle causal knowledge and to learn CBNs.

This chapter proposes an overview of basic definitions of classical Bayesian networks

over the first three sections. Using this background, Section 2.4 describes what exactly can be represented by a CBN and which queries we can resolve with. Finally, section 2.5 focuses on the approaches used to learn such causal networks.

2.2 Notations and definitions

This section gives some notations and basic definitions needed in the rest of this report.

2.2.1 Notations

Let V=X_1, X_2,...,X_n be a finite set on n discrete variables. A variable is denoted by an upper case letter (e.g. X, Y, X_i) and a state or value of that variable by the same lower-case letter (e.g. x, y, x_i). A set of variables is denoted by a bold-face capitalized letter (e.g. $\mathbf{X}$, $\mathbf{Y}$) and the corresponding bold-face lower case letter (e.g. $\mathbf{x}$, $\mathbf{y}$) denotes n assignments or states for each variable in a given set. Calligraphic letters (e.g. $\mathcal{B}$, $\mathcal{G}$, $\mathcal{E}$) denotes statistical models both parametrized and not.

Let E be a part of V$\times$ V, then $\mathcal{G}$=(V,E) is said to be a **graph** on V and E corresponds to the set of **edges** connecting some pairs of nodes in V. If the edges in E are oriented then they are called **arcs** and $\mathcal{G}$=(V,E) is said to be a **directed graph**.

2.2.2 Definitions

- For each arc X1X2, the node X1 is called its origin and X2 its end.
- In an arc X1X2, the node X1 is the **parent** of X2 and the node X2 is the **child** of X1.
- A **root** is a node with no parents.
- A **leaf** is a node with no children.
- Two nodes linked by an edge are said to be **adjacent**.

Figure 2.1: a v-structure

- A **path** in a directed graph is a sequence of nodes from one node to another using the arcs.
- A **directed path** from X1 to Xn in a DAG $\mathcal{G}$ is a sequence of directed edges X1→X2...→Xn. The directed path is a cycle if X1=Xn, i.e. it begins and ends at the same variable.
- A **semi directed path** from X1 to Xn in a partially acyclic directed graph is a path from X1 to Xn such that each edge is either undirected or directed away from X1.
- A **chain** in a graph is a sequence of nodes from one node to another using the edges.
- A **cycle** is a path visiting each node once and having the same first and last node.
- A **DAG** is a Directed Acyclic (without cycles) Graph.

  For any node X_i∈V corresponds the following sets:

  - $Pa(X_i)$: the parent set of X_i.
  - $Desc(X_i)$: the descendent set of X_i.
  - $Ch(X_i)$: the child set of X_i.
  - $Anc(X_i)$: the ancestor (i.e. non-descendent) set of X_i.
  - $Ne_U(A)$: the neighbor set of X_i.
- A **clique** is a set of vertices, such that for every two vertices, there exists an edge connecting the two. Alternatively, a clique is a graph in which every vertex is connected to every other vertex in the graph.
- A **v-structure** is defined as an ordered triple of nodes (X1, X2, X3) such that $\mathcal{G}$ contains the directed edges X1→X2 and X3→X2 and X1 and X3 are not adjecent in $\mathcal{G}$ (See Figure 2.1).

Figure 2.2: (a) a singly connected DAG, (b) a multiply connected DAG

Figure 2.3: (a) example of DAG, (b) the skeleton relative to the DAG and (c) an example of a PDAG.

- A **Singly Connected DAG or polytree** is a DAG which contains no loops, in this case the graph obtained by dropping the directions of the links is a tree (See Figure 2.2(a)).

- A **Multiply Connected DAG** is a DAG which can contains loops (See Figure 2.2(b)).

- A **Partially Directed Acyclic Graph (PDAG)** is a graph which contains both directed and undirected edges, with no directed cycles in its directed subgraphs.

- A **Complete Partially Directed Acyclic Graph (CPDAG)** is a graph consisting of directed and undirected edges. An edge X1$\longrightarrow$X2 is directed if all BNs in the equivalence class have an edge X1$\longrightarrow$X2 with the same directionality. An edge is undirected if some have X1$\longrightarrow$X2 and some X1$\longleftarrow$X2.

- The **skeleton** of any DAG is its underlying undirected graph obtained by transforming the set of directed edges into a set of undirected ones that preserves the same adjacencies.

2.3 Bayesian networks

2.3.1 Definition

Over the last decade, BNs have become a popular representation for encoding uncertain expert knowledge in expert systems [23]. Formally, a BN over a set of variables V consists of two components:

- ***graphical component*** composed of directed acyclic graph (DAG) $\mathcal{G}$ reflecting the causal relations relative to the modeled domain. Bayesian networks encode the conditional independence assumption exposed in Property (4).

- ***numerical component*** consisting in a quantification of different links in the DAG by a conditional probability distribution of each node X_i in the context of its parents ($Pa(X_i)$). Such conditional probabilities should respect the following normalization constraints for each variable X_i:

  – if $Pa(X_i)$=$\emptyset$ (i.e. X_i is a root), then the a priori probability relative to X_i should satisfy: $\sum_{x_i} P(\mathrm{x}_i)$=1

  – if $Pa(X_i)\neq\emptyset$, then the relative conditional probability relative to X_i in the context of its parents $Pa(X_i)$ should satisfy: $\sum_{x_i} P(x_i|\ \mathrm{Pa}(x_i))$=1

Property 4 ***(Markov assumption)*** *Each variable X_i in $\mathcal{G}$ is independent of its nondescendants given its parents.*

At this step, the Markov assumption allows us to obtain a factorized representation of the unique joint probability encoded by a Bayesian network via the following chain rule:

$$P(X_1, X_2, ..., X_n) = \prod_{i=1..n} P(x_i \mid Pa(X_i)). \tag{2.1}$$

Figure 2.4: Example of DAG (V = X1, X2, X3)

Example 5 *Given the Bayesian network represented by the DAG in the figure 2.4 and the a priori and conditional probabilities in tables, the joint probability distribution is defined by:*
$P\ (x_1\ ,\ x_2\ ,\ x_3) = P(x_2 \mid x_1\ ,\ x_3) \times P(x_1) \times P(x_3) \quad \forall\ x_1,\ x_2,\ x_3.$
For instance, $P\ (x_1$=true , x_2=true , x_3=true) = 0.4 × 0.2 × 0.5 = 0.04

Table 2.1: A priori probabilities

x_1	P(x_1)	x_3	P(x_3)
$true$	0.4	$true$	0.2
$false$	0.6	$false$	0.8

Table 2.2: Conditional probabilities

x_2	x_1	x_3	$P(x_2 \mid x_1 \wedge x_3)$	x_2	x_1	x_3	$P(x_2 \mid x_1 \wedge x_3)$
$true$	$true$	$true$	0.5	$false$	$true$	$true$	0.5
$true$	$true$	$false$	0.1	$false$	$true$	$false$	0.9
$true$	$false$	$true$	0.3	$false$	$false$	$true$	0.7
$true$	$false$	$false$	0.2	$false$	$false$	$false$	0.8

2.3.2 The d-Separation criterion

Let us consider three disjoint sets of variables X,Y and Z, which are represented as nodes in a directed acyclic graph G. To test whether X is independent of Y given Z in any distribution compatible with G, we need to test wheteher the nodes corresponding to variables Z "block" all paths from nodes in X to nodes in Y. By blocking we mean stopping the flow of information (or of dependency) between the variables that are connected by such paths[41].

In order to define the d-separation criterion, we need first to present the three basic connection structures between variables.

- **Converging or Head-to-head connection:** If we know the value of X_3, then X_1 can give a supplementary on X_2 and vice versa, so X_1 and X_2 are *independent* in the context of X_3 (see Figure 2.5(a)) .
- **Serial or Head-to-tail connection :** If we know the value of X_3, then X_1 will be irrelevant to X_3 and consequently to X_2, so X_1 and X_2 are *independent* in the context of X_3 (see Figure 2.5(b)) .
- **Diverging or Tail-to-tail connection:** If we know the value of X_3, then X_1 will be irrelevant to X_2 and vice versa, so X_1 and X_2 are *independent* in the context of

X_3(see Figure 2.5(c)) .

Figure 2.5: The three basic connection structures

Hence, the d-separation criterion can be defined as follows [40]:

Definition 6 *d-separation:*

I) Let $\mathcal{G}$ = (V,E) be a DAG and let X, Y and Z be three distinct variables in V. Then, X and Z are said to be d-separated by Y , if for all paths between X and Z, there is an intermediate variable Y such that either:

i) the connection is serial or diverging and the state of Y is known, or

ii) the connection is converging and neither the state of Y , nor any of its descendants, are known.

II) Let $\mathcal{G}$ = (V,E) be a DAG and let X, Y and Z be three disjoint subsets of V. Then, X and Z are said to be d-separated by Y, if for every variable $X_i \in X$ and $Z_i \in Z$, X_i and Z_i are d-separated by Y.

Example 7 *Let us consider the DAG represented by figure 2.6*

- *The nodes X_1 and X_4 are d-separated given X_3 because the path between X_1 and X_4 contains a converging connection in X_3 and the state of X_3 is unknown. In the other case, if X_3 was measured, X_1 and X_4 will be d-connected given X_3.*

- *However X_2 and X_5 are d-connected given X_4 because $X_2 \rightarrow X_4 \rightarrow X_5$ is a serial connection and the state of X_4 is unknown. If X_4 was measured, the path between X_2 and X_5 will be blocked by X_4.*

Figure 2.6: Example of DAG (V = X_1, X_2, X_3, X_4, X_5)

2.3.3 Inference with Bayesian networks

One of the main tasks insured by BNs is to determine how the incoming observations, relative to some variables, affect the probabilities of the remaining variables. This process of computing the posterior distribution of variables given evidence is called *probabilistic inference* or *propagation*. Mathematically, the task were boiled down to the computation of P(y | x), where X is a set of observations and Y is a set of variables that are deemed important for prediction or diagnosis.

Given a joint distribution P, the computation of P(y | x) is based on the Bayes's rule:

$$P(y \mid x) = \frac{\sum sP(y, x, s)}{\sum y, sP(y, x, s)} \tag{2.2}$$

where S stands for the set of all variables excluding X and Y. The challenge, however, lies in performing these computations efficiently. Although such inference techniques are not essential to our discussion of causality.

Many algorithms were developed to perform propagation in BNs, but we can partition them into two classes:

- **Exact algorithms:** We will distinguish two types of exact methods. The first is the fundamental algorithm for exact probability propagation in graphs with no cycle, which was proposed by Kim and Pearl [29] and Pearl [40, 39]. In this algorithm, the impact of each new piece of evidence is viewed as a perturbation that propagates in parallel through the network, via a message passing mechanism, between neighboring variables. For the second method, the JLO algorithm (JLO for Jensen, Lauritzen & Olesen) [26, 27] is well suited for the propagation in multiply connected graphs. The principle of this algorithm is to transform the initial graph into a junction tree of clusters of variables. The messages will be transmitted between clusters allowing the computation of marginal distributions.

- **Approximate algorithms:** There is certain number of approximate algorithms based on stochastic methods such MCMC (Markov Chain Monte Carlo) or Gibbs

Sampler [38].

2.3.4 Learning Bayesian networks

Before evoking two major families of approaches, we will begin by reminding the frame in which we are working. So, learning the structure of a baysien network from data means finding a graph which is a P-map of an independence model associated with a probability distribution. Thus, it is necessary to be certain about the existence of a such P-map (fidelity) and to know well all the variables (causal sufficiency).

The major problem when learning BNs from observational data is that we can not depict causal relationships, we only study correlations between variables. In the causal interpretation, this correlation of variables implies an unresolved causal structure, which means that if two variables X_i and X_j are correlated they will have one of the following underlying causal structures [48]:

- X_i is a cause of X_j
- X_j is a cause of X_i
- There is a common cause X_i of X_j and X_k

These three possibilities are depicted graphically in Figure 2.7. The third possibility occurs if there is latent variables that influence the system.

Figure 2.7: When there is a correlation between X_i and X_j, one of these three causal structures hold.

Generally, when learning BNs, an important property known as the Markov Equivalence is usually taken into consideration We considered that two Bayesian networks structures G1 and G2 are said to be equivalent, if they can be used to represent the same set of probability distributions.

Figure 2.8: Markov equivalence

More formally, Chickering [7] defines the Markov equivalence as follows:

Definition 8 *Two DAGs G1 and G2 are equivalent if for every BN B1=(G1,Θ1), there exists a BN B2=(G2,Θ2) such taht B1 and B2 define the same probability distribution, and vice versa.*

Example 9 *If we consider the example of figure 2.8, the decomposition of the joint probability distribution for respectively the networks (a) and (b) is as follows:*
$P(X1, X2, X3)_a = P(X1 \mid X2) \times P(X2 \mid X3) \times P(X3)$.
$P(X1, X2, X3)_b = P(X1) \times P(X2 \mid X1) \times P(X3 \mid X2)$.
However,
$P(X1,X2,X3)_b = P(X1) \times [P(X1|X2) \times P(X2)/ P(X1)] \times P(X2|X3) \times [P(X2|X3) \times P(X3)/P(X2)]$
$= P(X1|X2) \times P(X2) \times [P(X2|X3) \times P(X3)/P(X2)]$
$= P(X1|X2) \times P(X2|X3 \times P(X3) = P(X1,X2,X3)_a$.
Thus, we demonstrate that the networks (a) and (b) are equivalents. Similarly, the network (c) is equivalent to (a) and (b). Only network (d), which represents a v-structure, is not equivalent to the three others.

Moreover, Verma and Pearl [42] propose the following definition which provides a graphical criterion for determining the equivalence of two DAGs:

Definition 10 *Two DAGs are equivalent if and only if they have the same skeletons (i.e the undirected graph resulting from ignoring the directionality of every edge) and the same v-structures (i.e ordered triples of nodes (X1,X2,X3) such that* $X1 \longrightarrow X2$, $X3 \longrightarrow X2$ *and X1 and X3 are not adjacent.*

Several researches and studies were focused on learning Bayesian networks from observational data. There are essetially two main approaches for learning the structure of a Bayesian network:

Constraint-based approach

This first series of structure learning approaches, often called search under constraints, arises from works of both rival teams, Pearl & Verma for IC and IC* algorithms [41, 42], Spirtes, Glymour & Scheines [49] for the SGS, PC and FCI and recently the BN-PC algorithm of Cheng & all [4]. All these algorithms are based on the same principle:

- *Initialization*: Construct a complete undirected graph containing the relations between variables.
- *Skeleton discovery*: Use the conditional independencies (or dependencies) entailed from data to remove edges.
- *Edge orientation*: Detect the V-strctures.
- *Edge orientation, edge propagation*: Based on the already oriented edges, apply some orientation rules called PC rules to infer new arcs until no more edges can be oriented.

The main clause of all these methods lies in the determination from data of the relations of conditional independences between two variables conditionally to a set of variables.

Figure 2.9: ASIA theoretical network

Figure 2.10: Run example of PC

Example 11 *In what follows, we will detail the Predictive Causation (PC) algorithm [49], since it will be used in the next chapter.*

To more understand the working of PC algorithm, let us consider the famous theoretical network "ASIA" given in Figure 2.9. First of all, the initialization starts with a complete undirected network, as illustrated by Figure 2.10(a). The skeleton discovery phase consists in performing conditional independence tests. When a conditional independence between two variables is found, the corresponding edge can be removed. Assuming that we test independences concerning X_1 *first, we find that* $X_1 \perp X_2$ *and so the edge* $X_1 - X_2$ *is removed*

(See Figure 2.10(b)). Let us note that Figures 2.10(b), (c) and (d) illustate the process of deleting edges correponding respectively to conditional independence of order 0, 1 and 2. After performing (conditional) independence tests, the next step will be the edge orientation, in which we try to find the v-structures. Here, the triple $X_3 - X_6 - X_4$ *will be oriented as* $X_3 \rightarrow X_6 \leftarrow X_4$*, giving us Figure 2.10(e). Finally, Figure 2.10(f) is the result of applying the orientation rules to infer the direction of edges. It is clear in this example that conditional independence tests did not succeed to find three edges (*$X_1 \rightarrow X_3$*,* $X_6 \rightarrow X_7$ *and* $X_6 \rightarrow X_8$*). In the other hand, if the edge* $X_6 - X_7$ *were found, the edge orientation step in the PC algorithm will force its orientation to* $X_6 \rightarrow X_7$ *to not create new v-structure* $X_3 \rightarrow X_6 \leftarrow X_7$*.*

Algorithm 2.1: PC algorithm [49]

begin

1) *Initialization*:
   G is complete undirected graph on the vertex set V.

2) *Skeleton discovery*:
   n=0;
   repeat
   repeat
   select an ordered pair of variables X_i and X_j that are adjacent in G such that $Ne(X_i)\backslash X_j$ has cardinality $\geq n$, and a subset S of $Ne(X_i)\backslash X_j$ of cardinality n, and if X_i and X_j are independent given S delete edge $X_i - X_j$ from G and record S in $Sepset(X_i, X_j)$ and $Sepset(X_j, X_i)$

   until *all ordered pairs of adjacent variables X_i and X_j such that $Ne(X_i)\backslash X_j$ has cardinality $\geq n$ and all subsets S of $Ne(X_i)\backslash X_j$ of cardinality n have been tested for independence;*
   n=n+1;

   until *for each ordered pair of adjacent vertices X_i, X_j, $Ne(X_i)\backslash X_j$ is of cardinality $< n$*

3) *Edge orientation: v-structure discovery*
   for *each triple of vertices X_i, X_j, X_k such that the pair X_i, X_j and the pair X_j, X_k are each adjacent in G but the pair X_i, X_k are not adjacent in G* **do**
   orient $X_i - X_j - X_k$ as $X_i \rightarrow X_j \leftarrow X_k$
   if and only if X_j is not in $Sepset(X_i, X_k)$

4) *Edge orientation: inferred edges*
   repeat
   PC-Rules to come here

   until *no more edges can be oriented*

end

Score-based approach

Contrary to the first family of methods which tried to find conditional independences between variables, the following approaches go either look for the structures which maximize a certain score, look for the best structures and combine their results.

The score-based algorithms assign a score to each possible network based on how well the network represents the data and how complex the network is. Typically, a scoring metric consists of measuring how well the samples $\mathcal{D}$ from a distribution can be modeled by the network $\mathcal{G}$.

One of the commonly used scores are the *Bayesian information criterion* (BIC) and *minimum description length* (MDL), defined as follows:

$$Score_{BIC}(\mathcal{G},\mathcal{D})=\log \mathrm{L}(\mathcal{D}|\theta^{ML},\mathcal{G})\text{-}\tfrac{1}{2}\mathrm{Dim}(\mathcal{G})\log \mathrm{N}$$

$$Score_{MDL}(\mathcal{G},\mathcal{D})=\log \mathrm{L}(\mathcal{D}|\theta^{ML},\mathcal{G})\text{-}|A_G|\log \mathrm{N} \text{ - } \mathrm{c.Dim}(\mathcal{G})$$

where N is the number of samples, c the number of bits necessary to stock every unique parameter, $|A_G|$ the number of arcs and $Dim(G)$ is the number of free parameters in the graph in the graph G and θ^{ML} the set of parameters $P(X_i|\Pi_i)$ found by *Maximum Likelihood* (ML) estimation which estimates the probability of an event $P(x_i|\pi_i)$ by the frequency of this event in the data.

Here the major problem is how to cover the space B relative to bayesien networks, since the number of possible graphs grows exponentionally with the number of nodes [45]. As an exhaustive search is impossible to make, the proposed algorithms work on a reduced

space in which they make a greedy search. Within the most famous algorithms based on the score we can mention: K2 [10], Maximum Weight Spanning Tree (MWST) [9], Greedy Equivalence Search (GES) [5], Greedy search (GS) [6].

2.4 Causal Bayesian networks

In order to provide a causal interpretation for Bayesian networks, a causal extension appears, with specific properties and assumptions [41].

2.4.1 Definitions and properties

Definition 12 ***(Causal Bayesian networks)*** *A causal Bayesian network denoted by CBN, is a Bayesian network in which each directed edge represents an autonomous causal relation.*

Causal Bayesian networks provide a convenient framework for causal modelling and reasoning as they have a stricter interpretation of the meaning of edges than usual Bayesian network. In fact, every link between two variables represents a causal mechanism. A Causal Bayesian Network is defined as a Bayesian network that respects the following central properties:

Property 13 ***(Causal Markov assumption)*** *A node is independent of its distant causes given just its direct causes.*

Property 14 ***(Causal Sufficiency assumption)*** *There exist no common unobserved (also known as hidden or latent) variables in the domain that are parent of one or more observed variables of the domain. This assumption avoids the confounding factor problem.*

Property 15 ***(Causal Faithfulness assumption)*** *A Bayesian network graph G and a probability distribution P are faithful if and only if every one and all independence relations valid in P are those entailed by the Markov assumption on G.*

2.4.2 The "do" operator

In Pearl framework, the notification of external intervention (also known as manipulation, experiment, or action) on causal models, is expressed by the do operator. Definition The effect of an action "do(X=x)" in a causal model corresponds to a minimal perturbation of the existing system that sets the variable X to the value x. The distinction between observation (see) and manipulation (do) in causal analysis is expressed as follow:

- Conditional probability that variable Y = y when we see that X = x is noted: P(y | see(x)) = P(Y = y | X = x) = P(y | x)
- Conditional probability that variable Y =y when we do (i.e. set) X = x is noted: P(Y = y | do(X = x)) = P(y | do(x)) = $P_x(y)$

2.4.3 Conditioning Vs manipulating

Conditioning corresponds to mapping a probability distribution into a new distribution in response to finding out more information about the state of the world (or seeing).

Manipulating corresponds to mapping a probability distribution into a new probability distribution in response to changing the state of the world in a specified way (or doing, in the terminology of Pearl [41]).

To illustrate these two notions, let us consider the following example [49].

Example 16 *Consider a population of flashlights, each of which has working batteries and light bulbs, and a switch that turns the light on when it is in the on position and turns the light off when it is in the off position. Lets note that Switch can take on the value on or off, and Light can take on the value on or off.*

We will consider that:

- *P(Switch=On)=1/2*
- *P(Switch=Off)=1/2*

- *P(Light=On)=1/2*
- *P(Light=Off)=1/2*

The joint distribution relative to this population is the following:

- *P(Switch=On,Light=On)=1/2.*
- *P(Switch=On,Light=Off)=0.*
- *P(Switch=Off,Light=On)=0.*
- *P(Switch=Off,Light=Off)=1/2.*

Thus, given a randomly chosen flashlight, the probability that the bulb is on is 1/2. However, if semeone observes that a flashlight has a switch in the off position and don't have any idea about the light; in this case, the probability of the light being off, conditional on the switch being off, is just the probability of the light being off in the subpopulation in which the switch is off;

$$P(Light = off/Switch = off) = \frac{P(Light = off, Switch = off)}{P(Switch = off)} = 1.$$

Similarly, the probability of the switch being off, conditional on the light being off, is just the probability of the switch being off in the subpopulation in which the light is off;

$$P(Switch = off/Light = off) = \frac{P(Light = off, Switch = off)}{P(Light = off)} = 1.$$

So an important feature of conditioning is that each conditional distribution is completely determined by the joint distribution (except when conditioning on an event that has the probablity 0).

In contrast to conditioning, a manipulated probability distribution is not usually a distribution in a subprobablity of an existing population but is a distribution in a population formed by externally forcing a value on a variable in the system. That's why now we will manipulate the light to off. Of course, the resulting probability distribution depends on how

we manipulated Light to off. Suppose that we manipulate Light to off by unscrewing the light bulb, we obtain:

$$P(Switch = off/do(Light = off)) = P(Switch = off) = 1/2.$$

Hence, P(Switch=off | do(Light=off)) ≠ P(Switch=off | Light=off)
In this case, the manipulation is said to be an "'ideal manipulation"' of Light because an external cause was introduced (the unscrewing of the light bulb) that was a direct cause of Light and was not a direct cause of any other variable in the system.

On the other hand, if we manipulated Light to off by pressing the Switch to off by pressing the Switch to off, then of course the probability that Switch is off after the manipulation is equal to 1. That's why it will not be an ideal manipulation.

This illustrates two key features of manipulations. The first is that in some cases, the manipulated probability is equal to the conditional probability (e.g., P(Light=off | do(Switch=off))=P(Light=off | Switch=off)), and in other cases, the manipulated probability is not equal to the conditional probability (e.g., P(Switch=off | do(Light=off))≠P(Switch=off | Light=off). In this example, conditioning on Light=off raised the probability of Switch=off, but manipulating Light to off didi not change the probability of Switch=off. In general, if conditioning on the value of a variable X raises the probability of a given event, manipulating X to the same value may raise, lower, or leave the same the probability of a given event.

The second key feature of manipulations is that even though Light=on if and only if Switch=on in the original population, the joint distributions taht resulted from manipulating the values of Switch and Light were different.

In contrast to conditioning, the results of manipulating depend on more than the joint probability distribution, they depend on the causal relationships between variables. The reason that manipulating the switch position changed the status of the light is that the switch position is a cause of the status of the light; the reason that manipulating the light condition

did not change the switch position is that the status of the light is not a cause of the switch position.

Thus, discovering the causal relations between variables is a necessary step to correctly inferring the results of manipulations.

2.5 Learning CBNs

In the following section, we will present the studies that have been performed to learn CBN from observational and experimental data.

2.5.1 Active learning for CBN structure

The active learning is a score-based approach to learn the structure of causal networks without latent variables from perfect experimental data. In the literature, there are two formal frameworks for active learning, namely, Tong and Koller approach [52] and the utility approach developed by Murphy [37].

These techniques propose to perform experiments based on:

- the current belief about the structure,
- the causal information that will be gained by an experiment.

The belief is modeled by $P(G|D^i)$, a probability distribution over the set of DAGs given the data seen so far. They update this belief after each experiment and then reiterate the process. Since the space of DAGs is super exponential in the number of nodes, an approximation is needed for $P(G|D^i)$.

By assuming causal sufficiency and faithful distibution, Tong and Koller [52] consider an active learner that is allowed to conduct experiments (i.e. intervenes in the domain by setting the values of certain variables). They assume that there are a number of query variables that can be experimented on after which the influence on all other variables is

measured.

An intervention query, denoted by Q=q corresponds to an intervention performed on a subset of nodes Q by clamping their values to q. In order to choose the optimal experiment they introduce a utility function, the loss-function, based on the uncertainty of the direction of an edge, to help indicate which experiment gives the most information. Using the results of their experiments they update the distribution over the possible networks and network parameters. Since it is impossible to do this for the entire set of DAGs they use an approximation based on the ordering of the variables proposed by [18].

Murphy [37] proposed a technique which resembles the one of [52] but different approximations are used to overcome working in the space of DAGs. [37] used MCMC to approximate the belief state $P(G|D^i)$ and importance sampling to calculate the expected utility.

2.5.2 Learning CBN from mixture of observational and experimental data

Cooper and Yoo [12] proposed another score-based method which can learn the structure from an arbitrary mixture of imperfect observational and experimental data. It is assumed that the performed experiments are structural. A closed-form Bayesian scoring metric was derived that can be used in this context: the metric takes into account whether the data is from observations or from experiments and adapts the score likewise. The new scoring metric is an adaptation of the one proposed by [11] and [24] for observational data alone.

2.5.3 Decision theoretic approach for learning CBNs

Recently, Meganck et al. [35] proposed a greedy approach to learn CBNs from perfect observational and experimental data (See Figure 2.11). Two main phases can be distinguished:

Figure 2.11: Main description of the MyCaDo approach [35]

- First phase: Used to learn the CPDAG from observational dataset using traditional structure learning techniques such as the ones proposed in Section 2.3.4.

- Second phase: Also known as the MyCaDo (My Causal DiscOvery) algorithm. As input the MyCaDo algorithm needs a CPDAG and proceeds using three steps. Initially the choice concerns which experiment to perform and hence also which variables will be altered and measured. Next in order, the actual experiment should be performed, and at the end the results of the experiment will be analyzed. The results of the experiments will direct a number of edges in the partially directed structure. If there are still some edges that are undirected, the process will be re-iterated. When all edges are directed, the process is accomplished and the correct CBN modeling the original data is given.

1) **Choice of experiment:**
Meganck et al. [35] appeal to elements from decision theory to decide which experiment should be performed at each step. Generally, depending on the values of some input parameters, a decision problem uses a utility function to choose the appropriate actions that will lead to the best result. This choice is commonly based on some decision criteria such as least risk or optimistic estimation.

• *Utility function*: The utility function $U(.)$ proposed in [35] is a function of three variables: $gain(exp)$, $cost(exp)$, $cost(meas)$, respectively the gained information, the cost of performing an experiment and the cost of measuring other variables. The only restriction that is placed on this utility function is that it is proportional to $gain(exp)$ and inversely proportional to the sum of $cost(exp)$ and $cost(meas)$. The general formula is expressed by:

$$U(A_{X_i}) = \frac{\alpha gain(A_{X_i})}{\beta cost(A_{X_i}) + \gamma cost(M_{X_i})} \tag{2.3}$$

where A_{X_i} and M_{X_i} denote, respectively, performing an action (i.e. experiment) at X_i and measuring the neighboring variables.

α, β and γ are measures of importance for every term.

Borchani et al. [3] have recently proposed an algorithm in which they define the utility of performing an experiment at X_i in function of the number of undirected neighbors $Ne_U(X_i)$ (e.g. nodes that are connected to X_i by an undirected edge) and the neighbors of $Ne_U(X_i)$.

• *Gain of an experiment*: Bofore all else, Meganck et al. [35] assume that when performing an experiment on X_i, it is possible to measure all neighboring variables $Ne(X_i)$. Accordingly, they can direct all edges connecting X_i and $Ne(X_i)$ as a result of the experiment. This means that the gain, $gain(A_{X_i})$, is based entirely on the number of variables $Ne_U(X_i)$ that are connected to X_i by an undirected edge, which will be directed according to the result of the experiment. However, it is possible that directing one edge can infer direction of other edges, as seen in the final phase of the PC-algorithm in Algorithm 2.1. We can take into account the possibility of inferred edges in $gain(A_{X_i})$.

Note that the amount of edges of which the direction can be inferred after performing an experiment is entirely based on the instantiation of the undirected edges connected to the variable being experimented on. Instantiating an undirected edge is assigning a direction to it, so for instance if we have an edge $X_i - X_j$, then $X_i \rightarrow X_j$ and $X_i \leftarrow X_j$ are the two possible instantiations of that edge. If we denote $inst(A_{X_i})$ as the set of instantiation of the undirected edges connected to X_i, the number of inferred edges based on $inst(A_{X_i})$ will be noted as $\#inferred(inst(A_{X_i}))$.

Occasionally, the CPDAG can be constitued of two partially directed substructures that are not linked in any way, in this case every subgraph can not be influenced by performing an experiment in the others, when the CPDAG is correct.

• *Cost of experiment and measurement*: The first line of defense against exper-

imentation goes typically like the following: "Doing an experiment would be incredibly expensive". Generally, there is a systematic budget for the expenditure of a fixed resource. Additionally to the amount of money required, the cost of an experiment includes also the time and the amount of space needed to perform it. Therefore, we should perform meaningful experiments that fit available budget. In many cases, certain experiments can not be performed, either because of ethical reasons (e.g. cloning humans) or simply because it is impossible to do so (e.g. provoking a seism). These types of experiments will be assigned a cost value of infinity (∞), in other terms, the gain of performing such an experiment will be 0.

• *Decision criteria*: In order to select the experiment to perform, Meganck et al. proposed a number of decision criteria. By maximizing any of these criteria, they can be sure to choose the optimal experiment at each stage. Depending on the type of situation in which to perform the experiments it might be advantageous to choose a specific criterion. The *maximax* decision criterion favors the choice that might lead to the most directed edges. For this reason, it is considered as an optimistic criterion.

$$X_{best} = \underset{X_i}{argmax} \left(\frac{\#Ne_U(X_i) + \underset{inst(A_{X_i})}{max}(\#inferred(inst(A_{X_i})))}{cost(A_{X_i}) + cost(M_{X_i})} \right) \tag{2.4}$$

The second term in the numerator represents the maximum number of inferred edges by any of the instantiations of the directions of the undirected edges connected to X_i. The *maximin* decision criterion is a pessimistic one, since it consider the least number of possible inferred edges that can be found after performing an experiment on a variable X_i.

$$X_{best} = \underset{X_i}{argmax} \left(\frac{\#Ne_U(X_i) + \underset{inst(A_{X_i})}{min}(\#inferred(inst(A_{X_i})))}{cost(A_{X_i}) + cost(M_{X_i})} \right) \tag{2.5}$$

The *expected utility* is based on a distribution of edge directions, using the members in the equivalence of the graph under study, of any instantiation.

This mean that we perform the experiment on the variable X_{best} with: eq:maximax, eq:maximin, eq:equivalencetry

$$X_{best} = \underset{X_i}{argmax} \left(\frac{\#Ne_U(X_i) + \sum\limits_{inst(A_{X_i})} \#inferred(inst(A_{X_i}))P_{eq}(inst(A_{X_i}))}{cost(A_{X_i}) + cost(M_{X_i})} \right) \tag{2.6}$$

where $P_{eq}(inst(A_{X_i}))$ is the number of times a certain instantiation is present in the equivalence class divided by the number of members in that class.

2) **Detailed learning algorithm:**

Algorithm 2.2: Adaptive learning of CBN from observational and experimental data.

begin

(a) Apply a BN learning structure on D_O to obtain CPDAG G.

(b) Compute $U(A_{X_i})$ for each node X_i and find $X_{best} = argmax(U(A_{X_i}))$ using Equations 2.4, 2.5 and 2.6.

(c) Perform an experiment on variable X_{best}.

(d) Instantiate the orientation of all the undirected edges connected to X_{best}
for *all* $X_j \in Ne_U(X_{best})$
do
if *distribution of* X_j *changed because of experiment*
then
orient $X_{best} - X_j$ as $X_{best} \rightarrow X_j$
else
orient $X_{best} - X_j$ as $X_{best} \leftarrow X_j$

(e) Infer edges (Orientation rules of PC, see Algorithm 2.1)
repeat
if $X_i \rightarrow X_j$ *and* X_j *and* X_k *are adjacent,* X_k *and* X_i *are not and there is no arrow into* X_k **then**
orient $X_j - X_k$ as $X_j \rightarrow X_k$.
if *there is a directed path from* X_i *to* X_j *and an edge between* X_i *and* X_j **then**
orient $X_i - X_j$ as $X_i \rightarrow X_j$.
until *no more edges can be oriented*

(f) Return to Step (2) until all links are directed.

(g) Return CBN G.

end

The adaptive algorithm described in Algorithm 2.2 summarizes the different phases discussed above. Remember that previously we considered the case where we have mul-

Figure 2.12: Toy CBN (left) and the corresponding CPDAG (right)

tiple subgraphs that are not connected by undirected links. All the substructures will be trated separately by applying multiple instances of the algorithm.

3) **Toy example:**
The following toy example illustrates the working mechanim of the different criteria described above. As shown in Figure 2.12, the CBN modeling the system is given on the left and the correct CPDAG, which represents the input of the algorithm is shown on the right.

To simplify the calculations in this example, the following cost assignments will be considered:

$Cost(A_{X_i}) = 1, \forall i$
$Cost(M_{X_i}) = \#Ne_U(X_i)$

The toy example treats the only case when we perform an action on X_2. In Figure 2.13 we give an overview of all possible instantiations of the edges $X_2 - Ne_U(X_2)$ and the corresponding inferred edges. Henceforth, it is possible to calculate the utility for each decision criteria.

The cost is equal for all decision criteria, namely:
$Cost(A_{X_2}) + cost(M_{X_2}) = 1 + 2 = 3.$

The complexity of the calculation of the utilities for each decision criteria is dependent on two phases. First, we need to search among all instantiations of $X_i - Ne_U(X_i)$ for the set of possible instantiations in the particular PDAG. Second, we need to check which edge directions can be inferred based on this instantiation. Both these phases are simple searches for patterns in a graph and can be performed efficiently. All decision criteria operate in the same way and hence have the same type of complexity.

Maximax: The maximum number of inferred edges is 1, so the utility for Maximax is

Figure 2.13: All possible instantiations for $Xi - Ne_U(Xi)$, the possible structures compatible with this instantiation and the result of inferring edges.

given by:

$$U(X_2) = \left(\frac{\#Ne_U(X_2) + \max_{inst(A_{X_2})} (\#inferred(inst(A_{X_2})))}{cost(A_{X_2}) + cost(M_{X_2})} \right) = \frac{3+1}{1+2} = \frac{4}{3}$$

Maximin: The worst number of edges inferred is 0, so the utility for Maximin is given by:

$$U(X_2) = \left(\frac{\#Ne_U(X_2) + \min_{inst(A_{X_2})} (\#inferred(inst(A_{X_2})))}{cost(A_{X_2}) + cost(M_{X)}} \right) = \frac{3+0}{1+2} = \frac{3}{3} = 1$$

Expected utility: To use the Expected utility criterion, given in Equation 2.6, we need to calculate the number of graphs in the equivalence class of the PDAGs we encounter.

As shown in Figure 2.13, there are 28 DAGs in the equivalence class of our example (all the possible structures that can be inferred for all the instantiations). So the Expected utility for X_2 is given by:

$$\begin{aligned} U(X_2) &= \left(\frac{\#Ne_U(X_2) + \sum_{inst(A_{X_2})} \#inferred(inst(A_{X_2}))P_{eq}(inst(A_{X_2}))}{cost(A_{X_2}) + cost(M_{X_2})} \right) \\ &= \frac{3 + 1\frac{3}{28} + 1\frac{3}{28} + 1\frac{3}{28} + 1\frac{3}{28} + 0\frac{1}{7} + 0\frac{1}{7} + 0\frac{1}{7} + 0\frac{1}{7}}{1+2} = \frac{8}{7} \end{aligned}$$

2.6 Conclusion

In this chapter we showed how probabilistic Bayesian networks can be extended to represent causal relationships between variables in a system.

Furthermore we showed that this new semantic interpretation of the arcs in a CBN allows to perform causal inference, which is the calculation of the effect of performing an intervention on the system.

The causal interpretation of the edges has as a consequence that learning the structure of a CBN no longer amounts to finding a member of the equivalence class but finding the complete causal structure. We gave an overview of state-of-the art algorithms for handling this task.

In the next chapter we will introduce our approach for learning the complete structure of a CBN.

Chapter 3

New Strategy for Iterative Causal Discovery & Visualization

3.1 Introduction

In the previous chapter, we have shown that in order to learn CBNs, the choice of variables to experiment on can be crucial when the number of experiments is restricted. Therefore, every additional knowledge can improve the causal discovery.

In many cases, available ontologies provide high level knowledge for the same domain under study. The recourse to ontologies is due essentially to their ability to capture the semantics of domain expertise. Hence, a lot of ontological solutions have been implemented in several real applications in different areas as natural language translation [30], medicine [19], electronic commerce [32] and bioinformatics [1]. Therefore, the semantical knowledge contained in the ontology can turn out of a big utility to improve causal discovery.

This chapter is devoted to introduce a new strategy to integrate ontological knowledge for more efficient causal discovery. Henceforth, we propose adapted tools to enrich the visualization of large causal networks (i.e. generally composed of hundreds or thousands nodes), based on the ontological knowledge extracted from ontologies.

Section 3.2 presents the characteristics of the ontology which motivates the development of this approach. Section 3.3 defines the Semantical Inertia Criterion, the ontological knowledge integration term used by our approach, as well as its required properties. Section 3.4 discusses the second contribution of our works, consisting in an enriched visualization of causal models integrating the power of ontological knowledge. Finally, we present a toy example illustrating the whole process.

3.2 Basics on Ontologies

The metaphisical studies on ontologies starts since the Aristotle time. Ontology as a branch of philosophy is the science that studies the constituents of reality and answers questions for being and existence in the universe. The term "ontology" is derived from the Greek word "onto" (i.e. being) and "logia" (i.e. written or spoken discourse). Since the artificial intelligence considers that what "exists" is that which can be represented, the notion of ontology have become a principal topic of interest in computer science.

There are different definitions in the literature of what should be an ontology. The most notorious was given by Tom Gruber [21], stipulating that an ontology is an *explicit* specification of a *conceptualisation*. The "conceptualization", here, refers to an abstract model of some phenomenon having real by identifying its relevant concepts. The word "explicit" means that all concepts used and the constraints on their use are explicitely defined.

Moreover, this definition is often extended by 3 additional conditions, namely, an ontology is an explicit, *formal* specification of a *shared* conceptualisation od a *domain of interest*. The use of "formal" is justified since the ontology should be machine readable. "Shared" reflects common understanding that reaches across people and application systems.

Therefore, a number of ontology definition languages (e.g. OIL [1], DAML [2], DMAL+OIL [3], RDF [4]/RDFS [5] and OWL [6], etc.) have been developed over the past few years.

On the other hand the reference to a domain of interest indicates that the goal of ontologies is not modelling the whole world but rather modelling just parts which are relevant to the task at hand.

Formally, an ontology is characterized by three components:

- *concepts* (classes, objects or categories)
- *characteristics* (attributes, properties)
- *relations* (generalization, specialization)

The relations in an ontology are either taxonomic (e.g. is-a, part-of) or non-taxonomic (i.e. user defined). The taxonomic ones are the commonly used relations. In the simple case, the ontology takes the form of a tree or hierarchy representing concept taxonomy. Formally:

Definition 17 *A* ***Concept Taxonomy*** $H = (C, E, Rt)$ *is a directed acyclic graph where,* $C = \{c_1, c_2, ..., c_n\}$ *is the set of all concepts in the hierarchy,* E *is the set of all subsumption links (is-a) and* Rt *is the unique root of this DAG. The primary structuring element of concept taxonomy is the subsumption relationship, which supposed that if the concept* c_i *is a child of concept* c_j *then all properties of* c_j *are also properties of* c_i *and we say that* c_j *subsumes* c_i *.*

Regarding any concept taxonomy hierarchy, we can give the following notations:

- $pths(c_i, c_j)$: the set of paths between the concepts c_i and c_j, where $i \neq j$,

[1] http://www.ontoknowledge.org/oil/
[2] http://www.daml.org/
[3] http://www.daml.org/2001/03/daml+oil-index
[4] http://www.w3.org/RDF/
[5] http://www.w3.org/TR/2004/REC-rdf-schema-20040210/
[6] http://www.w3.org/2001/sw/WebOnt/

- $len_e(e)$: the length in number of edges of the path e,
- $mscs(c_i, c_j)$: the most specific common subsumer of c_i and c_j, where $i \neq j$.

We will interest now to the basic primitives of an ontology which are concepts and relations. Among the set of possible relations some of them are not used systematically.

Example 18 *Figure 3.1 shows an example of Gene Ontology (GO). This ontology is an is-a tree one, where the leaf nodes (X1, X2, X3, X4, X5) are genes. Their direct subsumers (F1, F2, F3) represent the biological functions i.e. every set of genes in a GO share a very specified function. Also, the biological functions are subsumed by their biological super-functions (SF1, SF2). Note that we can use the term of generalization as shown in figure 3.1 to refer the subsuming process. For this case of is-a tree, only one path can be find between two different concepts. To more illustrate this purpose, let us consider the two concepts X1 and SF2. pths(X1, SF2) is the set of the edges X1-F1, F1-SF1, SF1-RT and RT-SF2. Consequently the length in number of edges of the corresponding path will be equal to four. Moreover, we specify that mscs can be applied to more than two concepts, as follows: mscs(X1, X2, X3)=mscs(mscs(X1, X2), X3)=mscs(F1,X3)=SF1.*

Figure 3.1: Example of hierarchy representing a taxonomy of concepts.

Recently, several works highlighted the importance of evaluating the strength of the semantic links inside domain ontologies. We can distinguish three major classes of semantic measures, namely *semantic relatedness*, *semantic similarity* and *semantic distance*, evaluating, respectively, the resemblance, the closeness and the disaffection between two concepts.

The *semantic similarity* represents a special case of *semantic relatedness*. For instance, if we consider the two concepts *wind turbine* and *wind*, they would be more closely related than, for example the pair *wind turbine* and *solar panel*. However the latter concepts are more similar. Therefore, all pairs of concepts with a high semantic similarity value (i.e. high resemblance) have a high semantic relatedness value whereas the inverse is not necessarily true. In the other hand, the semantic distance is an inverse notion to the semantic

relatedness.

The major approaches of measuring semantic distance are *Rada et al.'s distance* [43], *Sussna's distance* [50] and *Jiang and Conrath's distance* [28]. For the semantic similarity, we find *Leacock and Chodorow's similarity* [31], *Wu and Palmer's similarity* [53] and *Lin similarity* [33], while for semantic relatedness, the most used one is *Hirst and St Onge's relatedness*. See [2] for a comparative study of these measures.

In what follows, we will focus on semantic distances and in particular on the classical *Rada et al.'s distance* [43], which can be easily replaced by any other semantic distance. This distance is based on the shortest path between the nodes corresponding to the items being compared such that the shorter the path from one node to another, the more similar they are. Thus, given multiple paths between two concepts, we should take the length of the shortest one. It, also, supposed that all the taxonomic links between two adjacent concepts have the same value. Formally, given two concepts c_i and c_j the Rada et al.'s distance is defined by:

$$dist_{rmbb}(c_i, c_j) = \min_{p \in pths(c_i, c_j)} len_e(p) \tag{3.1}$$

Consequently, we will compute a distance matrix, giving us all distances between all pairs of concepts.

Example 19 *In figure 3.1, every concept has at least one generalization's path back to the top node. The full Rada et al.'s distance matrix concerning the concept taxonomy is presented in Table 3.1. For instance,* $dist_{rmbb}(SF1, X2) = 2$ *since the shortest path between the two concepts SF1 and X2 is composed of the two edges SF1-F1 and F1-X2.*

Table 3.1: Rada et al.'s distance matrix between the nodes given in Figure.3.1

	Rt	SF1	SF2	F1	F2	F3	X1	X2	X3	X4	X5
Rt	0	1	1	2	2	2	3	3	3	3	3
SF1	1	0	2	1	1	3	2	2	2	4	2
SF2	1	2	0	3	3	1	4	4	4	2	4
F1	2	1	3	0	2	4	1	1	3	5	3
F2	2	1	3	2	0	4	3	3	1	5	1
F3	2	3	1	4	4	0	5	5	5	1	5
X1	3	2	4	1	3	5	0	2	4	6	4
X2	3	2	4	1	3	5	2	0	4	6	4
X3	3	2	4	3	1	5	4	4	0	6	2
X4	3	4	2	5	5	1	6	6	6	0	6
X5	3	2	4	3	1	5	4	4	2	6	0

3.3 Integrating ontological knowledge for causal discovery

Through reviewing the literature, there are less works that have tried to combine the power of BNs and ontologies. Most of them focus on extending the current ontology representation formalism with uncertain reasoning [15, 54], which become crucial in various tasks such as domain modeling, ontology reasoning and concept mapping between ontologies. In this section, we present our approach for integrating ontological knowledge for iterative causal discovery.

As already detailed in Chapter 2, CBNs are used to model domains under uncertainty. In many cases, available ontologies provide consensual representation of the same domain and a full description of the knowledge model. Basically, ontologies attempt to find the complete set of concepts covering any domain as well as the relationships between them. Such domain knowledge can be efficiently used to enrich the learning process of CBNs and optimize the causal discovery.

In order to illustrate this idea we propose the SemCaDo (SEMantical CAusal DiscOvery) algorithm which is an extension of the MyCaDo algorithm (See algorithm 2.2) used to learn causal structures with experiments. SemCaDo algorithm aims to integrate ontological knowledge, extracted using Rada et al. semantic distance calculation, to improve causal discovery and visualization.

We make the assumption that we consider a concept taxonomy hierarchy, in which the leaves are also the nodes of the Bayesian network. Moreover, the causal relations we want to discover and model with our CBN did not exist in the corresponding ontology.

In what follows let us assume that after performing an experiment on a variable X_i, we can measure all neighboring variables $Ne_U(X_i)$.

Semantical inertia

Our utility function $U(.)$ is an extension of Equation 2.3, by generalizing the first term $\#Ne_U(Xi)$ and replacing it by the semantical inertia, denoted by $SemIn(Ne_U(X_i))$.

This notion will enable us to integrate our ontological knowledge extracted by calculating semantic distances from ontologies and guide the choice of the best experiment. In many situations, we can have a node X_i with a high number of neighbors but all those neighbors are very close semantically. This implies that we will have, exactly, two alternatives (i.e. X_i is the direct cause of all his neighbors or the inverse).

For instance, in bio-informatics, biologists would discover more causal relations between biological functions. Considering a neighborhood which is very close semantically will reduce the number of functions under study and consequently the informative contribution of the experiment would be very low. Properly speaking, in the case of distant neighbors, there will be more cause-to-effect relations between biological functions to find.

Figure 3.2: All possible instantiations for $X2 - Ne_U(X2)$

Such causal discoveries have an important scientific contribution in the bio-informatic field.

The semantical inertia will be as follows:

$$SemIn(M) = \frac{\sum_{X_i \in M} \min_{p \in pths(X_i,\ mscs(M))} (len_e(p))}{card(M)} \tag{3.2}$$

Let us consider the CPDAG in figure 3.7 in order to illustrate our approach and compare it to the original MyCaDo. We will also use the concept taxonomy hierarchy presented in figure 3.1. Suppose that we will perform an action on X_2. Figure 3.2 summarizes all possible instantiations of the edges X_i-$Ne(Xi)$.

As stated before, for the reason that the 2 neighbors X_3 and X_5 share the same function in the corresponding ontology, the following edges $X_2 - X_3$ and $X_2 - X_5$ will have the same direction. In this case, we will have three instantiations whereas, by applying MyCaDo we will find exactly six. Consequently, we reduced considerably the set of possible instantiations in the particular PDAG (i.e. CPDAG with some oriented edges).

Here the most specific commun subsumer of $\mathrm{Ne}_U(X_2)$ $=\{\mathrm{X}_1, X_3, X_5\}$ is the concept SF1. According to the ontology in figure 3.1 and table 3.1, the semantical inertia of the nodes X_1, X_3 and X_5 is as follows:

$$\begin{aligned} SemIn(X_1, X_3, X_5) &= \frac{\sum_{X_j \in Ne_U(X_2)} \min_{p \in pths(X_j,\ mscs(Ne_U(X_2))} (len_e(p))}{\#Ne_U} \\ &= \frac{2+2+2}{3} = 2 \end{aligned}$$

The semantical inertia presents three major characteristics. At the first glance, where all undirected neighboring nodes belong to the same biological function, the semantical inertia of the neighborhood will be equal to one. Secondly, the sementical inertia depends on the

number of undirected neighbors. For example, if we eliminate X_1 from the neighborhood of X_2, SemIn will automatically decrease.

$$SemIn(X_3, X_5) = \frac{1+1}{2} = 1$$

And, finally, the more the neighboring variables are distant according to the ontology, the more the semantical inertia will be important and the utility maximized. Here, if we replace the node X_3 by X_4, which is more distant from the rest of neighbors of X_2, SemIn will increase considerably.

$$SemIn(X_1, X_4, X_5) = \frac{3+3+3}{3} = 3$$

It is clear now that the semantical inertia represent a generalization of $\#Ne_U(Xi)$ and introducing it in the utility function will allow a better choice of experiments, based on ontological knowledge. With such method, we can focus the causal discovery on relations between distant concepts.

3.4 Integrating ontological knowledge for causal graph visualization

Visualizing large networks of hundreds or thousands nodes is a real challenging task. It is essentially due to the limitation of the screen, the huge number of nodes and edges and the limitations of human visual abilities.

In this context, our approach can turn out of big utility to improve actual visual tools. The main purpose was to propose an enriched visualization of causal models by integrating the power of semantical knowledge. Henceforth, the Rada et al.'s distance matrix can implement different graph drawing algorithms among which MultiDimentional Scalling (MDS) and Force Directed Placement [14, 13].

Generally, we use these algorithms to position the graph nodes in two or three dimentional space. They proceed by assigning forces amongst the set of edges and the set of

Figure 3.3: (a) An example of graph; (b) An example of applying a Force Placement Algorithm to the initial graph.

nodes. Roughly speaking, the entire graph will be simulated to a physical system where edges are springs and nodes are electrically charged particles. A set of forces will be applied to the nodes pulling them closer together or pushing them further apart until the system comes to an equilibrium state. At that moment, we obtain an adjusted graph in which the nodes are positioned referring to the matrix distance calculated above. Similarly, in [44], Ranwez et al. used the ontological distance measures to propose an alternative information visualization on conceptual maps.

In the other hand, we propose to adapt the size of the nodes to the utility function of MyCaDo algorithm. This method will allow biologists to determine the ideal node to make interventions.

While judicious use of semantical distances and node's size can help considerably the understanding of the causal bayesian network, we can go farther in the visualization by adding a zoom-in/zoom-out function that allows one to visualize either the global structure of the graph or just, smaller components reduced to more general concepts in the ontology.

To give such useful representation, we need to decompose the data into groups and if necessary to decompose each group into sub-groups and so on. In that work, we will adopt the concept taxonomy hierarchy presented in Figure 3.1 as a cluster tree. This concept taxonomy represents a meaningful specific decomposition of the domain under study, in order to obtain a *clustered graph* [16].

Definition 20 *A Clustered Graph W=(G,T) consists of a graph G and a rooted tree T such that the leaves of T are exactly the vertices of G. Each node v of T represents a cluster*

of vertices of G that are the leaves of the subtree of T rooted at v. The root of T represents the whole graph G.

Figure 3.4: an example of tree hierarchy

Figure 3.5: Two graph visualizations correponding to level 3 (a) and level 1 (b) of the tree hierarchy presented in Figure 3.4

Figure 3.4 presents an example of tree hierarchy with *height*=3. Here, the notion of height or level is very important since it will enable us to draw clustered graphs using a multilevel Force-Directed Graph Drawing algorithm. This means that for each level in the tree hierarchy, an intelligent placement and a force-directed algorithm are performed. An illustration is given by Figure 3.5, showing how we can obtain a top-bottom multilevel representation of the causal graph. For instance, we can pass from visualizing a graph reduced to genes, in our case, to an abstraction of biological functions or even biological super-functions.

All those visualization attributes can lead to different valuable informations about the causal bayesian network. Moreover, we will be able to investigate further into the adapted structure of the network.

Figure 3.6: Main description of the SemCaDo approach

Since the enriched visualization can offer assistance to the domain experts, we can change the SemCaDo algorithm into semi-automatic method, combining the use of optimal solution of utility and visualization of causal graph. Roughly speaking, in every iteration, an enriched visualization of the causal graph will be proposed to the expert which can choose manually the node to manipulate or to be content with the SemCaDo experiment choice (See Figure 3.6). Consequently, we can ease considerably the depiction of causality.

3.5 Toy Example

Figure 3.7: An example of CPDAG

Let us reconsider the same toy example proposed in the previous chapter in order to illustrate our approach and to compare it to the original MyCaDo. We will also use the ontology presented in figure 3.1 and keep the same cost assignments:

$Cost(A_{X_i}) = 1, \forall i$

$Cost(M_{X_i}) = \#Ne_U(X_i)$

After performing an experiment on X_2, we summarize, in the first column of figure 3.8 all possible instantiations of the edges X_i-$Ne_U(Xi)$. It is clear now that the following edges $X_2 - X_3$ and $X_2 - X_5$ have the same direction because the two neighbors X_3 and X_5 share the same function in the corresponding ontology. In this case, we will obtain four instantiations whereas, by applying MyCaDo we will find exactly seven. Consequently, we reduced considerably the set of possible instantiations in the particular PDAG. For each instantiation, we present all possible structures(i.e. edges that can be inferred after applying PC rules) and the result of edge inference.

Figure 3.8: All possible instantiations for $Xi - Ne_U(Xi)$, the possible structures compatible with this instantiation and the result of inferring edges.

Here the most specific commun subsumer of the nodes X_1, X_3 and X_5 is the concept SF1. According to the ontology in figure 3.1 and table 3.1, the semantical inertia of the nodes X_1, X_3 and X_5 will be equal to two (See Section 3.3 for more calculation details).

Thus, the three decision criteria will give us the following results:

- **Maximax:** the maximum number of inferred edges is 3, so the utility for Maximax is given by:

$$
\begin{aligned}
U(X_2) &= \frac{SemIn + \max\limits_{inst(A_{X_2})}(\#inferred(inst(A_{X_2})))}{cost(A_{X_2}) + cost(M_{X_2})} \\
&= \frac{2+3}{1+2} = \frac{5}{3}
\end{aligned}
$$

- **Maximin:** the worst number of edges inferred is 1 such that the utility for Maximin will be:

$$
\begin{aligned}
U(X_2) &= \frac{SemIn + \min\limits_{inst(A_{X_2})}(\#inferred(inst(A_{X_2})))}{cost(A_{X_2}) + cost(M_{X_2})} \\
&= \frac{2+1}{1+2} = 1
\end{aligned}
$$

- **Expected utility:** to use the Expected utility criterion, we need to calculate the number of graphs in the equivalence class of the PDAGs we encounter.

  As shown in figure 2.13, there are 7 DAGs in the equivalence class of our example (all the possible structures that can be inferred for all the instantiations). So the Expected utility for X_2 is given by:

$$
\begin{aligned}
U(X_2) &= \frac{SemIn + \sum\limits_{inst(A_{X_2})} \#inferred(inst(A_{X_2}))P_{eq}(inst(A_{X_2}))}{cost(A_{X_2}) + cost(M_{X_2})} \\
&= \frac{2 + (3\frac{1}{7} + 1\frac{3}{7} + 1\frac{3}{7})}{1+2} = \frac{23}{21}
\end{aligned}
$$

  Since the decision criteria equations are not normalized, the results obtained can go beyond the value of one.

Figure 3.9: an example of graph correponding to level 3 (a) and level 2 (b) of the tree hierarchy presented in Figure 3.4

In figure 3.9.a, we show (right) an enriched visualization of the original CPDAG. The visualization tools enabled us to obtain a graph with real aesthetic properties (semantical distances, node's size...). The main advantage of this technique is the interactive aspect since the user can select different levels to visualize either the global structure of the graph or just, smaller components reduced to more general concepts in the ontology. Figure 3.9.b shows a top level visualization of the corresponding CPDAG.

3.6 Conclusion

In this chapter, we proposed the SemCaDo algorithm to learn CBNs from a mixture of perfect experimental and observational data. We described how we can integrate the high-level knowledge, extracted from the ontology, in order to improve the decision theoretic approach. Our basic motivation is to take into consideration the power of the semantical knowledge to optimize the causal discovery and offer a semi-automatic solution, combining the use of optimal utility and CBNs visualization.

In next chapter, an experimental study is perfomed in order to show the distinct improvements carried out by SemCaDo algorithm in comparison with the MyCaDo algorithm [35].

Chapter 4

Implementation and Experimentations

4.1 Introduction

In this chapter, we present the experimentation results relative to our learning algorithm SemCaDo. Indeed, using developments of Chapters 3, we propose an implementation based on the API *ProbT* [1] which is a set of high-performance modules developed in C++ which allows the creation of Bayesian models. We also used the ontology editor *protégé 4.0 beta* [2] to extract the ontological knowledge from the OWL files and we integrate the *Tulip* [3] software to enable faster analysis by the end users.

This chapter is composed of two parts: The first part presents the principle implementation tools and details our experimental plan. Then, the second part presents the experimental results and gives a comparative analysis between SemCaDo and MyCaDo algorithms.

[1] http://www.probayes.com/
[2] http://protege.stanford.edu/
[3] http://tulip.labri.fr/

4.2 Programming Tools

To implement the main tasks of the SemCaDo algorithm, we used the **ProbT**, which is a set of high-performance modules developed in C++ language. This API aims to provide a programming tool that facilitates the creation of Bayesian models and their use for learning and inference either in exact or approximate way. Once the Bayesian model constructed, the user can "ask" all possible inference query of the form P(Variables | Evidence) using the ProBT-Engine, which execute all required probabilitic calculus. The ProBT Library API is an Application Program Interface available in C++ and runs in the most common systems: Linux/Unix, Windows, and MacOS X.

We also used the ontology editor and knowledge acquisition system **protégé 4.0 beta** to extract the ontological knowledge from the concept taxonomies. Protégé is based on Java, is extensible, and provides a plug-and-play environment that makes it a flexible base for rapid prototyping and application development. Ontologies developed with Protégé can be exported into a variety of formats including RDF, OWL and XML Schema.

The model visualization will be ensured by **Tulip software**, a framework, created by David Auber, providing automatic drawing and graph clustering, in which the 3 dimentional vizuliation tools enable faster analysis by the end users.

4.3 Implementation

To evaluate the theoretical results of our approach, we have developed several programs grouped by principal tasks. Figure 4.1 gives an overview of the SemCaDo implementation strategy.

Figure 4.1: CBN, CPDAG and Obseravational Data generation

4.3.1 Ontology & Random CPDAG Generation

Figure 4.2: An example of protégé output

First of all, we create our ontology using an ontology editor *protégé 4.0 beta* . An example of protégé output is given by Figure 4.2. We integrate the SemanticTab plugin in order to generate the Rada & al. Distance matrix, that will be used to calculate decision criteria of SemCaDo algorithm.

Since we make the assumption that all variables in the CBN should be modelled by the ontology, the exact number of nodes of the causal graph will be fixed given the concept taxonomy. For example, in Figure 4.1, the CBN variables correspond to the leaves of the concept taxonomy given in Figure 4.2.

In order to test our approach on a large set of CBNs, we implemented a random CBN generator providing the original structure G_O. Simultaneously, we randomly sample data sets records, based on the structure as well as the underlying probability distribution of the network. Then we implemented a DAG-TO-CPDAG function returning the principal input of the SemCaDo algorithm, as depicted in Figure 4.1.

As suggested by its name, the DAG-TO-CPDAG algorithm takes as input a BN structure (i.e. DAG), and outputs a CPDAG representation of the equivalence class to which that structure belongs. As defined by Chickering [7, 8], the first step of this algorithm is to determine a total ordering over the edges in the given DAG. This step is sketched by the following algorithm:

Algorithm 4.1: Order-Edges

```
begin
    Perform a topological sort on the nodes in 𝒢  i ← 0;
    while there are unordered edges in 𝒢 do
        Let Y be the lowest ordered node that has an unordered edge incident into it
        Let X be the highest ordered node for which X → Y is not ordered
        Label X → Y with order i
        i ← i + 1
end
```

Then, the DAG-TO-CPDAG algorithm labels all the edges in the DAG as either compelled or reversible, as shown in Algorithm .

Algorithm 4.2: Label-Edges

```
begin
    Order the edges in 𝒢 using Algorithm 4.1
    Label every edge in 𝒢 as unknown
    while there are edges labeled unknown in 𝒢 do
        stop ← 0 Let X → Y be the lowest ordered edge that is labeled unknown
        for every edge W → X labeled compelled do
            if W is not a parent of Y then
                Label X → Y and every edge incident into Y
                with compelled stop ← 1
            else
                Label W → Y with compelled
            if stop = 0 then
                if there exists an edge Z → Y such that Z ≠ X and Z is not a parent
                of X then
                    Label X → Y and all unknown edges incident into Y with com-
                    pelled
                else
                    Label X → Y and all unknown edges incident into Y with reversible
end
```

Given such a labelling, it is trivial, therefore, to construct the corresponding $CPDAG$

consisting in a directed edge for every compelled edge, and an undirected edge for every reversible edge, as outlined by the following algorithm:

Algorithm 4.3: DAG-TO-CPDAG

```
begin
    Label the edges in 𝒢 using Algorithm 4.2
    for every edge e in 𝒢 do
        if e is labeled compelled then
            e is directed in 𝒫^c and has the same orientation as in 𝒢
        else
            e is undirected in 𝒫^c
    Return 𝒫^c
end
```

When using the boost[4] library in Probt, only directed graphs or undirected graphs can be created and manipulated. Therefore, we tried to get around this problem by using the weights that can be fixed for each edge in the graph. In other words, we will create an undirected graph, in which we assign the weight of -1 to the reversible edges and try to manipulate those weights in next steps.

4.3.2 Interventional Data Sets Generation

Since real interventions are not really available, we will generate them in a synthetic manner by mutilating the original network. This means that to perform an experiment on X_i, we have to do a surgical intervention by cutting all incoming arcs on X_i as shown in Figure 4.3. Consequently, the experimented node will not be influenced by its parents.

Figure 4.3: An example of graph mutilation

The graph mutilation gives us a post-intervention structure G_E. All parameters besides the one for the variable experimented on (i.e. $P(X_i|\Pi_i)$) remain the same.

[4]http://www.boost.org/

In other words, we force X_i to take on random values and then sample the post-intervention distribution P_E for the variables in $Ne_U(X_i)$ using G_E to get our experimental data set D_E (See Figure 4.4).

Finally, a Chi-square adjustment test will be used to determine if the variable experimented on is the cause or the effect.

Figure 4.4: Experimental data generation from a known CBN

Example 21 *Let us consider a CPDAG composed of 2 binary variables (X and Y). We manipulate the first variable and we tried to get the correct direction of the link connecting it to its direct neighbor.*
Suppose that initially we have 100 observational data. The total number of observations where Y=y_1 (resp. Y=y_2) is equal to 30 (resp. 70). This means that P(Y=y_1)=30/100=0,3 and P(Y=y_1)=70/100=0,7.
The experimental data can be obtained by different operations of do on the experimented variable. We force X to take on one value (e.g. X=x_1) and we count the occurence number of Y=y_i in the generated experimental data in order to get $P(Y = y_i \mid do(X = x_1))$.
Note that the number of experimental data generated after the manipulation can be different from the number of initial observational data.
Here, with $do(X = x_1)$, we obtain 200 experimental data. The total number of experiments where Y=y_1 (resp. Y=y_2) is equal to 50 (resp. 150) and P(Y=y_1)=50/200=0,25 (resp. P(Y=y_1)=150/200=0,75).
We denote the total number of experimental data (resp. the number of experiments where Y=y_1 and the number of experiments where Y=y_2) by Nexp (resp. $Nexp(Y = y_1)$ and $Nexp(Y = y_2)$).
Theoretically, if X is not the direct cause of Y then P(Y=y_i) will be equal to $P(Y = y_i \mid do(X = x_1))$.
Practically, the previous equality is true for only infinite number of experimental data. This means that we should apply a Chi-square adjustment test to determine if there is significant association between Y before intervention and Y after every do(X=x_1).

Before proceeding, we fix the degree of freedom (DF) to Card(Y)-1 where Card(Y) represents Y cardinalities. Chi Square statistic will be equal to:
$\sum_i Nexp(Y = y_i)$-$(Nexp \times P(Y=y_i)))^2$ / $(Nexp \times P(Y=y_i))$
$=(50-60)^2/60+(150-140)^2/140 = 100/60+100/140 = 2{,}38$.
We now have our chi square statistic (χ^2=2,38) and our degree of freedom (DF =1). We use the Chi-Square Distribution Calculator to find $P(\chi^2 \geq 2{,}38)=0{,}12$.
Depending on the value of the significance level α, if the last result is close to 0 then the laws before and after manipulation are approximately the same. In our case, we are far from 0 then $P(Y = y_1) \neq P(Y = y_1 \mid do(X = x_1))$ and X is the direct cause of Y.
Generally, if at least one statistical test confirm that the 2 studied laws are differents then $P(Y = y_i) \neq P(Y = y_i \mid do(X = x_1))$ and X is the direct cause of Y.
On the other hand, if all statistical tests assert that there is a significant association then $P(Y = y_i) = P(Y = y_i \mid do(X = x_1))$ and X will be the effect of Y.

4.4 Complexity Analysis (MyCaDo Vs SemCaDo)

To calculate the decision criteria (Maximax, Maximin and Expected Utility), the MyCaDo algorithm needs to process all possible instantiations for X_i-$Ne_U(X_i)$. Those instantiations represent different combinations of directed edges between the studied variable and its neighborhood. The major problem with this approach is that we need to know the exact number of members in the equivalence class of the current PDAG.

The theoretical complexity of this task is $O(2^n)$, here n is the number of neighbors. So there is no doubt that the number of elements in the equivalence class grows exponentially when the number of neighboring variables increases.

Nevertheless, in our approach, we only focus on the number of ontological sets to which belong the neighboring variables of X_i. This implies that theoretical complexity will be equal to $O(2^k)$, where k is the number of sets. In most cases, the number of ontological sets is less than the number of neighbors. This means that we reduce both number of

instantiations and possible structures. This does not exclude that they can be equals when neighbors don't share any ontological set.

We also remember that the goal of SemCaDo is not to minimize the total number of experiments needed to completely direct the CPDAG, we would rather minimize the total cost of those experiments, maximize the total number of edges that will be directed by each experiment and guide the manipulation process to the graph variables with high semantical inertia. This will lead indirectly to a decrease in the number of interventions.

Eberhardt et al. [17] demonstrate that at most N-1 experiments (when N is the number of variables in the graph) allow us to determine all causal relations among the N Variables. Generally, this theoretical result is much higher than the amount of experiments needed to fully direct the CPDAG. Nonetheless we want to emphasize that in the worst case scenario, where there is a clique graph, the upper bound will be attained.

4.5 Experimental results

As stated above, the aim of integrating ontological knowledge in the SemCaDo algorithm is not to reduce the number of experiments needed to fully direct the CPDAG. we would rather guide the manipulation process to the graph variables with high semantical inertia.

Figure 4.5: The number of experiments needed given the number of nodes in the graph

In Figure 4.5, we plotted the average number of experiments needed to completely orient a DAG with a given number of nodes going from 0 to 30. The 3 curves correspond to the 3 decision criteria (i.e. Maximin, Maximax and Expected Utility). It is clear that the number of experiments needed is still related to the size of the graph.

Our main contribution consists of guiding the choice of experiments to the graph zones with high semantic inertia, according to Rada & al. Distance matrix. These interventions are usually very expensive to perform, therefore the choice of variables to experiment on can be vital when the number of experimentations is restricted.

4.6 Conclusion

In this chapter, we have detailed the main tasks of the SemCaDo algorithm.

We have, also, presented a comparative complexity analysis between our approach and the MyCaDo algorithm. The experimental study shows that SemCaDo algorithm guides the choice of best experiments to semantically rich variables, with almost better execution time.

Moreover, the second contribution of SemCaDo algorithm consisting in the enriched visualization of causal graphs can offer an assistance to the domain experts. This means that we can change SemCaDo algorithm into semi-automatic method, combining the use of optimal solution of utility and the use of the enriched visualization of causal graphs, in order to fully direct CBNs.

Conclusion

"I would rather discover one causal
relation than be King of Persia."
Democritus(460-370 B.C.)

Causal discovery is an immensely rich area of research. Many application fields such as medicine, biology, economics, and social science could benefit from the discovery of even one causal relation.

In this master thesis, we have presented and discussed a novel approach for integrating ontological knowledge for iterative causal discovery.

Roughly speaking, we extend the MyCaDo algorithm [35] by introducing the notion of semantical inertia. By supporting the assignment of costs to experiments and measurements, SemCaDo allows us to guide the choice of the best experiment in order to obtain the complete causal bayesian network.

Then we proposed an enriched visualization of the causal models, using the power of semantical knowledge extracted from ontologies.

We notice that summarized version of this dissertation [36] has been accepted for oral presentation in the Machine Learning and Visualization Workshop (Hammamet 2009) and will be published in the proceedings of in the 10th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2009) and publica-

tion in the proceedings.

Nevertheless, the proposed work is still subject to improvements. As future work, we will tend to investigate different horizons that we classify into three major research directions:

- Remove the perfect observational data requirement and propose an experiment strategy, in which the available observational data does not provide enough information to learn the correct CPDAG. To this end, we can extend the UnCaDo algorithm proposed by Meganck & al. [35] by integrating the ontological knowledge in the experimentation phase.
- Ignore the causal sufficiency assumption by extending the actual SemCaDo algorithm to deal with latent variables that can influence the system under study, using the results of MyCaDo++ algorithm [34].
- Consider the case of incomplete interventional data (i.e. experiments returning some missing values).

Bibliography

[1] M. Ashburner, C. A. Ball, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock. Gene ontology: tool for the unification of biology. the gene ontology consortium. pages 25–29, 2000.

[2] E. Blanchard, M. Harzallah, H. Briand, and P. Kuntz. A typology of ontology-based semantic measures. *In 2nd INTEROP-EMOI Open Workshop on Enterprise Models and Ontologies for Interoperability at the 17th Conference on Advanced Information Systems Engineering (CAISE'05)*, 160:407–412, 2005.

[3] H. Borchani, M. Chaouachi, and N. Ben Amor. Learning causal bayesian networks from incomplete observational data and interventions. *In proceedings of Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, 2007.

[4] J. Cheng, D. A. Bell, and W. Liu. Learning belief networks from data: An information theory based approach. *In Proceedings of the sixth ACM International Conference on Information and Knowledge Management CIKM*, pages 325–331, 1997.

[5] D. Chickering. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3:507–554, 2002.

[6] D. Chickering, D. Geiger, and D. Heckerman. Learning bayesian networks: Search methods and experimental results. *In Proceedings of Fifth Conference on Artificial Intelligence and Statistics*, pages 112–128, 1995.

[7] D. M. Chickering. A transformational characterization of equivelent bayesian networks. *In the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 87–98, 1995.

[8] D. M. Chickering. Learning equivalence classes of bayesian-network structures. *Journal of Machine Learning Reasearch*, pages 445–498, 2002.

[9] C. K. Chow and C. N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE trans.on Info.Theroy*, pages 462–467, 1968.

[10] G. Cooper and E. Hersovits. A bayesian method for the induction of probabilistic networks from data. *Machine Learning*, pages 309–347, 1992.

[11] G. F. Cooper and E. Herskovits. A bayesian method for the induction of probabilistic networks drom data. *Machine Learning*, pages 309–347, 1992.

[12] G. F. Cooper and C. Yoo. Causal discovery from a mixture of experimental and observational data. *In Proceedings of Uncertainty in Artificial Intelligence*, pages 116–125, 1999.

[13] M. Crampes, S. Ranwez, F. Velickovski, C. Mooney, and N. Mille. An integrated visual approach for music indexing and dynamic playlist composition. *in proceedings of 13th Annual Multimedia Computing and Networking (MMCN'06), San Jose, CA, US*, 2006.

[14] M. Crampes, S. Ranwez, J. Villerd, F. Velickovski, C. Mooney, A. Emery, and N. Mille. Concept maps for designing adaptive knowledge maps. *Concept Maps, A Special Issue of Information Visualization, Vol. 5,*, 2006.

[15] Z. Ding. *BayesOWL: A Probabilistic Framework for Semantic Web.* PhD thesis, University of Maryland, 2005.

[16] P. Eades and Q. W. Feng. Multilevel visualization of clustered graphs. *in proceedings of Graph Drawing, GD, number 1190*, pages 18–20, 1996.

[17] F. Eberhardt, C. Glymour, and R. Scheines. N-1 experiments suffice to determine the causal relations among n variables. *in Innovations in Machine Learning, D. Holmes and L. Jain (eds.), Theory and Applications Series: Studies in Fuzziness and Soft Computing, Vol. 194, Springer-Verlag. See also Technical Report CMU-PHIL-161*, 2005.

[18] N. Friedman and D. Koller. Being bayesian about network structure. *In proceedings of the 16th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 201–210, 2000.

[19] A. Gangemi, D. Pisanelli, and G. Steve. Ontology alignement: Experiences with medical terminologies. *In N. Guarino (ed.) Formal Ontology in Information Systems*, pages 163–178, 1998.

[20] I. Good. A tentative measure of probabilistic causation relevant to the philosophy of the law. *J. Statist. Comput. and Simulation 47*, 1993.

[21] T. R. Gruber. Towards principles for the design of ontologies used for knowledge sharing. *International Journal Human-Computer Studies 43*, pages 907–928, 1993.

[22] J. Y. Halpern and J. Pearl. Causes and explanations: A structural-model approachpart i: Causes. *In Proceedings of the Seventeenth Conference on Uncertainy in Artificial Intelligence, San Francisco*, pages 194–202, 2001.

[23] D. Heckerman. A tutorial on learning beyesian networks. *Technical Report MSR-TR-95-06, Microsoft Research*, 1995.

[24] D. Heckerman, D. Geiger, and M. Chickering. Learning bayesian networks: The combination of knowledge and statistical data. *Machine learning*, pages 197–243, 1995.

[25] C. H. Hennekens and J. E. Buring. *Epidemiology in Medicine*. Little Brown Co, 1987.

[26] F. V. Jensen. *Introduction to Bayesian networks*. UCL Press, University College, London, 1996.

[27] F. V. Jensen and F. Jensen. Optimal junction trees. *In proceedings of the tenth Conference on Uncertainty in Artificial Inteligence (UAI'94)*, pages 361–366, 1994.

[28] J. J. Jiang and D. W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. *In Proceedings of International Conference on Research in Computational Linguistics, Taiwan*, 1997.

[29] J. H. Kim and J. Pearl. A computational model for causal and diagnostic reasoning in inference systems. *In proceedings of International Joint Conference on Artificial Intelligence (IJCAI'83)*, pages 190–193, 1983.

[30] K. Knight and S. Luk. Building a large knowledge base for machine translation. *In Proceedings of American Association of Artificial Intelligence Conference (AAAI-94), Seattle, WA*, 1994.

[31] C. Leacock and M. Chodorow. Combining local context and wordnet similarity for word sense identification. *In Fellbaum C., ed.: WordNet: An electronic lexical database, MIT Press*, 1998.

[32] F. Lehmann. Machine-negotiated, ontology-based edi(electronic data interchange). *In Proceedings of CIKM-94 Workshop on Electronic Commerce*, 1995.

[33] D. Lin. An information-theoretic definition of similarity. *In Proceedings of the 15th International Conference on Machine Learning, Morgan Kaufmann*, pages 296–304, 1998.

[34] S. Meganck, P. Leray, S. Maes, and B. Manderick. Causal graphical models with latent variables: Learning and inference. *In Proceedings of the Ninth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty ECSQARU*, 2007.

[35] S. Meganck, P. Leray, and B. Manderick. Learning causal bayesian networks from observations and experiments: a decision theoretic approach. *In Proceedings of the Third International Conference MDAI, Lecture Notes in Artificial Intelligence*, pages 58–69, 2006.

[36] M. Ben Messaoud, P. Leray, and N. Ben Amor. Integrating ontological knowledge for iterative causal discovery. *In proceedings of Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, 2009.

[37] K. P. Murphy. Active learning of causal bayes net structure. *Technical report, Departement of Computer Science, UC Berkeley*, 2001.

[38] R. M. Neal. Probabilistic inference using markov chain monte carlo methods. 1993.

[39] J. Pearl. Fusion propagation and structuring in belief networks. *Artificial Intelligence*, pages 241–288, 1986.

[40] J. Pearl. Probabilistic reasoning in intelligent systems: Networks of plausible inference. *Morgan Kaufmann PublishersSan francisco (California)*, 1988.

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

[42] J. Pearl and T. S. Verma. A theory of inferred causation. *In Proceedings of Principles of Knowledge Presentation and Reasoning*, pages 441–452, 1991.

[43] R. Rada, H. Mili, E. Bicknell, and M. Blettner. Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man and Cybernetics*, 19:17–30, 1989.

[44] S. Ranwez, V. Ranwez, J. Villerd, and M. Crampes. Ontological distance measures for information visualization on conceptual maps. *In Proceedings of the OTM Workshops. LNCS 4278, Springer*, pages 1050–1061, 2006.

[45] R. W. Robinson. Counting unlabeled acyclic digraphs. in c. h. c. little, ed., combinatorial mathematics, lecture notes in mathematics, berlin: Springer. *In Lecture Notes in Computer Science, Springer*, 622:28–43, 1977.

[46] R. Scheines. Causation. *in New Dictionary of the History of Ideas*, 2004.

[47] R. Scheines. Causation, statistics and the law. *Journal of Law and Policy*, 16, 2008.

[48] B. Shipley. *Cause and correlation in biology.* Cambridge University Press, 2000.

[49] S. Spirtes, G. Glymour, and R. Scheines. *Causation, Prediction and Search.* MIT Press, 2001.

[50] M. Sussna. Word sense disambiguation for free-text indexing using a massive semantic network. *In Proceedings of the Second International Conference on Information and Knowledge Management*, pages 67–74, 1993.

[51] P. Thagard. Explaining disease: Correlations, causes, and mechanisms. *Minds and Machines*, pages 61–78, 1998.

[52] S. Tong and D. Koller. Active learning for structure in bayesian networks. *In proceedings of the National Conference on Artificial Intelligence, AAAI*, pages 567–573, 2002.

[53] Z. Wu and M. Palmer. Verb semantics and lexical selection. *In Proceedings of the 32nd Annual Meeting of the Associations for Computational Linguistics*, pages 133–138, 1994.

[54] Y. Yang and J. Calmet. Ontobayes: An ontology-driven uncertainty model. *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce Vol-1 (CIMCA-IAWTIC'05),*, pages 457–463, 2005.