SemCaDo: une approche pour la découverte de connaissances fortuites et l'évolution ontologique

Montassar Ben Messaoud

UNIVERSITÉ DE NANTES

Ecole Polytechnique de l'Université de Nantes

ÉCOLE DOCTORALE

« SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE MATHEMATIQUES »

N° attribué par la bibliothèque

Année 2012

SemCaDo: une approche pour la découverte de connaissances fortuites et l'évolution ontologique

SemCaDo: an approach for serendipitous causal discovery and ontology evolution

THÈSE DE DOCTORAT

Discipline : Informatique

Spécialité : Informatique

Présentée
et soutenue publiquement par

Montassar BEN MESSAOUD

Le 3 juillet 2012, devant le jury ci-dessous

Président :	Stéphane Loiseau	Professeur à l'Université d'Angers
Rapporteurs :	Céline Rouveirol	Professeur à l'Université Paris 13
	Jin Tian	Professeur à l'Iowa State University
Examinatrice :	Boutheina Ben Yaghlane	Maître de conférences à l'Université de Carthage

Directeur de thèse : Philippe Leray Professeur à l'Université de Nantes

Co-Directeur de thèse : Nahla Ben Amor Maître de conférences à l'Université de Tunis

ED : 503-162.

(Uniquement pour STIM et SPIGA)

Declaration of originality: I hereby declare that this thesis is based on my own work and has not been submitted for any other degree or professional qualification, except when otherwise stated. Much of the work presented here has appeared in previously published conference and workshop papers. The chapters discussing the SemCaDo approach and evaluating its performance on real data are based on [5, 6]. Where reference is made to the works of others, the extent to which that work has been used is indicated and duly acknowledged in the text and bibliography.

Montassar Ben Messaoud (July 6, 2012)

Acknowledgements

My time at both ISG'Tunis and Polytech'Nantes has been influenced and guided by a number of people to whom I am deeply indebted. Without their help, friendship and encouragement, this thesis would likely never have seen the light of day.

I would first like to express my gratitude to all members of the jury, Celine Rouveirol, Boutheina Ben Yaghlane, Jin Tian and Stéphane Loiseau 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 indebted to my professor Boutheina Ben Yaghlane who kindly accepted to have interesting discussions with me.

Special knowledge to Sourour Ammar, Amanullah Yasin and Raphael

Mourad for their generous support and help during the programming phase.

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.

Abbreviations & Acronyms

BN	Bayesian Network
CBN	Causal Bayesian Network
CPDAG	Complete Partially Directed Acyclic Graph
DAG	Directed Acyclic Graph
$D_{O,E}$	Observational/Experimental dataset
EM	Expectation Maximization
FCI	Fast Causal Inference
GES	Greedy Equivalence Search
GO	Gene Ontology
GRN	Gene Regulatory Network
GS	Greedy Search
IC, PC	Inductive/Predictive Causation
JPD	Joint Probability Distribution
MCMC	Monte Carlo Markov Chain
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
SGS	Spirtes, Glymour & Scheines

Abstract

With the rising need to reuse the existing domain knowledge when learning causal Bayesian networks, the ontologies can supply valuable semantic information to define explicit cause-to-effect relationships and make further interesting discoveries with the minimum expected cost and effort. This thesis studies the crossing-over between causal Bayesian networks and ontologies, establishes the main correspondences between their elements and develops a cyclic approach in which we make use of the two formalisms in an interchangeable way. The first direction involves the integration of semantic knowledge contained in the domain ontologies to anticipate the optimal choice of experimentations via a serendipitous causal discovery strategy. The semantic knowledge may contain some causal relations in addition to the strict hierarchical structure. So instead of repeating the efforts that have already been spent by the ontology developers and curators, we can reuse these causal relations by integrating them as prior knowledge when applying existing structure learning algorithms to induce partially directed causal graphs from pure observational data. To complete the full orientation of the causal network, we need to perform active interventions on the system under study. We therefore present a serendipitous decision-making strategy based on semantic distance calculus to guide the causal discovery process to investigate unexplored areas and conduct more informative experiments. The idea mainly arises from the fact that the semantically related concepts are generally the most extensively studied ones. For this purpose, we propose to supply issues for insight by favoring the experimentation on the more distant concepts according to the ontology subsumption hierarchy. The second complementary direction concerns an enrichment process by which it will be possible to reuse these causal discoveries, support the evolving character of the semantic background and make an ontology evolution. Extensive experimentations are conducted using the well-known Saccharomyces cerevisiae cell cycle microarray data and the Gene Ontology to show the merits of the SemcaDo approach in the biological field where microarray gene expression experiments are usually very expensive to perform, complex and time consuming.

Key-words : Causal Bayesian networks, ontologies, experimentations, serendipitous, causal discovery, ontology evolution.

Résumé

En réponse au besoin croissant de réutiliser les connaissances déjà existantes lors de l'apprentissage des réseaux bayésiens causaux, les connaissances sémantiques contenues dans les ontologies de domaine présentent une excellente alternative pour assister le processus de découverte causale avec le minimum de coût et d'effort. Dans ce contexte, la présente thèse s'intéresse plus particulièrement au crossing-over entre les réseaux bayésiens causaux et les ontologies et établit les bases théoriques d'une approche cyclique intégrant les deux formalismes de manière interchangeable. En premier lieu, on va intégrer les connaissances sémantiques contenues dans les ontologies de domaine pour anticiper les meilleures expérimentations au travers d'une stratégie fortuite (qui, comme son nom l'indique, mise sur l'imprévu pour dégager les résultats les plus impressionnants). En effet, les connaissances sémantiques peuvent inclure des relations causales en plus de la structure hiérarchique. Donc au lieu de refaire les mêmes efforts qui ont déjà été menés par les concepteurs et éditeurs d'ontologies, nous proposons de réutiliser les relations (sémantiquement) causales en les adoptant comme étant des connaissances à priori. Ces relations seront alors intégrées dans le processus d'apprentissage de structure (partiellement) causale à partir des données d'observation. Pour compléter l'orientation du graphe causal, nous serons en mesure d'intervenir activement sur le système étudié. Nous présentons également une stratégie décisionnelle basée sur le calcul de distances sémantiques pour guider le processus de découverte causale et s'engager davantage sur des pistes inexplorées. L'idée provient principalement du fait que les concepts les plus rapprochés sont souvent les plus étudiés. Pour cela, nous proposons de renforcer la capacité des ordinateurs à fournir des éclairs de perspicacité en favorisant les expérimentations au niveau des concepts les plus distants selon la structure hiérarchique. La seconde direction complémentaire concerne un procédé d'enrichissement par lequel il sera possible de réutiliser ces découvertes causales et soutenir le caractère évolutif de l'ontologie. Une étude expérimentale a été conduite en utilisant les données génomiques concernant Saccharomyces cerevisiae et l'Ontologie des Gènes pour montrer les potentialités de l'approche SemCaDo dans des domaines ou les expérimentations sont généralement très coûteuses, complexes et fastidieuses.

Mots-clés : Réseaux bayesiens causaux, ontologies, expérimentations, startégie fortuite, découvertes causales, évolution ontologique.

Contents

List of Figures

List of Tables

The only immediate utility of all sciences, is to teach us how to control and regulate future events by their causes.

David Hume (1748), An Enquiry concerning Human Understanding.

Chapter 1

Introduction

1.1 Research context

Debates continue to flourish over the most important interactions touching today's technology industries, climate changes, business solutions and many other aspects of our everyday life. What directly affects our health, immune system, metabolism, behavior and senses ? What mechanisms explain the planet's shape, its rotation and its gravitational field ? What about our purchasing power ?

Due to these frequent complex situations, the Machine Learning community has become increasingly aware of the need for developing approaches that unify statistical and relational methods for learning. In this context, the Probabilistic Relational Models [39], a range of Statistical Relational Learning formalisms, seem to be well placed to reason about uncertainty and provide relational structure representations. Because of their elegant way for dealing with variables as well as the relationships that hold amongst them, the Probabilistic Relational Models have been successfully applied for a wide variety of domains such as social network analysis, biological systems, pattern recognition and other domains that involve relational data.

Probabilistic Graphical Models [61] are a class of Probabilistic Relational Models that can represent rich dependency structures and capture the causal process by which the data was generated. Their popularity essentially comes from the fruitful marriage between graph theory and probability theory [58]. Depending on the specific nature of the pairwise interactions among variables, there are basically three popular classes of Probabilistic Graphical Models:

- Directed ones such as Bayesian networks [84, 85, 55] and causal bayesian networks [45, 88, 98] are popular alternatives in artificial intelligence and machine learning applications. These models are more consistent in revealing unidirectional causality.
- Undirected Markov networks [68, 85] are more adapted to statistical physics and computer vision. They are often used to capture the spatial correlation or mutual dependencies between random variables.
- Chain graphs [67] (hybrid graphs combining directed and undirected edges) are most useful when there are both causal-explanatory and symmetric association relations among variables, while Bayesian networks specifically deal with the former and Markov networks focus on the later.

In the remainder of this thesis, we will focus on causal bayesian networks since they are more consistent with our research context. The principle difference between Causal Bayesian Networks and standard ones lies in two key ways:

- The task of causal structure discovery need interventional data in cases where purely observational data is inadequate.
- In the causal extension, we move from probabilistic inference to causal one.

For this purpose, an experimentation phase must be conducted on certain variables to identify the true causal links connecting them to their neighborhood. However, experiments are often difficult to conduct, greedy in terms of resources, costly or even impossible. In this context, the aim of this thesis is to propose a decisional strategy for allowing more efficient causal discovery, where experiments are chosen with a great care.

On the other hand, it should be noted that most of the recent knowledge-based systems are supplemented and enhanced by structured background knowledge representation such as ontologies. At first blush, it seems that Bayesian networks and ontologies have almost nothing in common but this does not preclude that some recent studies have addressed some issues related to the integration of the two formalisms. This work also suggests a way to integrate ontological knowledge to support the causal discovery process in the causal bayesian networks and vice versa.

In support of this research perspective, steps have been taken to ensure a close cooperation between the LARODEC "Laboratoire de Recherche Opérationnelle et de Contrôle de Processus" and LINA "Laboratoire d'Informatique de Nantes Atlantiques". Through a partnership project, this thesis has been following a joint supervision Ph.D. student program from both laboratories.

Our contribution consists in a decisional causal learning method which is:

- *collaborative*, since exchanges have been established between the two main knowledge representation formalisms (causal Bayesian networks and ontologies).
- *iterative*, since the experimentation protocol requires several cycles.
- and *hybrid*, since it relies on data collected from benchmark datasets and causal prior knowledge.

1.2 Thesis overview

The structure of the thesis is organized around four intertwined topics, see Figure 1.1.

Chapter 2 reviews the scientific background and establishes the terminology required for discussing Causal Bayesian Networks, thus providing the basis for the subsequent chapters of this thesis. It starts off by reminding some of the basic notations and definitions that are commonly used in the (Causal) Bayesian Network literature. Moreover, it clarifies what differentiates a traditional Bayesian Network from causal one. Having established these basic facts, we then assess the role of experimentations when making the causal discovery process. The chapter closes with an overview of existing approaches for learning Causal Bayesian Networks.

Chapter 3 presents the second formalism that we used in our contribution. Initially, a brief overview on the structure, scope and application areas of ontologies is given. Next, we outline the ontology evolution issues and requirements. The chapter ends with a classification of the main contributions that attempt to combine Bayesian Networks and ontologies.

Chapter 4 gives the main correspondences that we made between the Bayesian Networks and the ontologies. It is then followed by the thesis contribution in which we identify ways to guide the causal discovery process meaningfully and accordingly, make ontology evolution.

Chapter 5 concludes the thesis by summarizing the major results that we obtained through simulations. In addition, the approach was successfully validated on a real system (S. cerevisiae cell cycle microarray data). We describe here the idea behind Gene Ontology and the manner in which we use it in the context of gene pathway discoveries. We conclude by identifying opportunities for future research.

1.3 Publications

The following parts of this work have previously been published in different international conferences:

- Chapter 4 was partially incorporated in the proceeding contributions for ECSQARU 2009 conference [5].

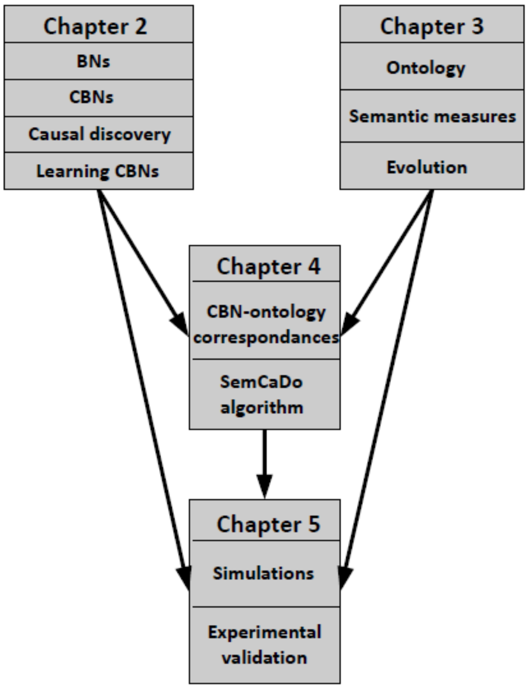


Figure 1.1: Interdependencies between chapters of the thesis

- A complete contribution, where we discuss the two interaction facets associated with coupling between Causal Bayesian Networks and ontologies, have been the object of two scientific publications in ARCOE 2011 [7] and ECSQARU 2011 [6].

This thesis will include selected passages from the above articles, mostly in paraphrased form.

Knowledge Discovery is the most desirable end-product of computing. Finding new phenomena or enhancing our knowledge about them has a greater long-range value than optimizing production processes or inventories, and is second only to task that preserve our world and our environment. It is not surprising that it is also one of the most difficult computing challenges to do well.

Gio Wiederhold, Standford University (1996)

Chapter 2

Causal Discovery & Bayesian Network -State of the art

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 [85].

Over the last few years, several researchers have proposed algorithms to learn Bayesian networks 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 random orientation of some edges to fully orient the partially directed structures.

In parallel, an extension of traditional Bayesian networks were introduced, where the semantics of edges are viewed as autonomous causal relations [88]. These causal Bayesian networks need, however, additional data to fully determine the true causal structure. More precisely, they extract causal knowledge from performing real experiments on the system under study. Several approaches and techniques have been developed to handle causal knowledge and to learn discrete causal Bayesian networks.

This chapter reviews basic definitions of classical Bayesian networks and causal discovery. Section 2.2 introduces some notations and definitions. Section 2.3 provides an overview of Bayesian networks. Using this background, Section 2.4 is relative to causal Bayesian networks. Finally, section 2.5 presents existing approaches used to learn these networks.

2.2 Notations and definitions

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

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). We use D_X=$\{x_1,...,x_n\}$ to denote the finite domain associated with each variable X_i and $|D_X|$ to fix the number of cardinalities. A set of variables is denoted by a bold-face capitalized letter (e.g. **X**, **Y**) and the corresponding bold-face lower case letter (e.g. **x**, **y**) denotes n assignments or states for each variable in a given set. Calligraphic letters (e.g. $\mathcal{G}$) denote statistical models.

2.2.2 Definitions

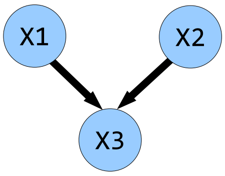


Figure 2.1: V-structure

- For each arc X_1→X_2, the node X_1 is called its origin and X_2 its end.

- In an arc $X_1 \rightarrow X_2$, the node X_1 is the **parent** of X_2 and the node X_2 is the **child** of X_1.
- 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**.
- A **path** in a directed graph is a sequence of nodes from one node to another using the arcs.
- A **directed path** from X_1 to X_n in a DAG $\mathcal{G}$ is a sequence of directed edges $X_1 \rightarrow X_2 ... \rightarrow X_n$. The directed path is a cycle if $X_1 = X_n$ (i.e. it begins and ends at the same variable).
- A **semi directed path** from X_1 to X_n in a partially acyclic directed graph is a path from X_1 to X_n such that each edge is either undirected or directed away from X_1.
- 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 (See Figure 2.3(a)).

For any node $X_i \in$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 set of X_i.
- $Ne_U(X_i)$: 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 subgraph in which every vertex is connected to every other vertex in the subgraph.

- 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 (See Figure 2.3(b)).

- A **v-structure** is defined as an ordered triple of nodes (X_1, X_2, X_3) such that $\mathcal{G}$ contains the directed edges $X_1 \rightarrow X_2$ and $X_3 \rightarrow X_2$ and X_1 and X_3 are not adjacent in $\mathcal{G}$ (See Figure 2.1).

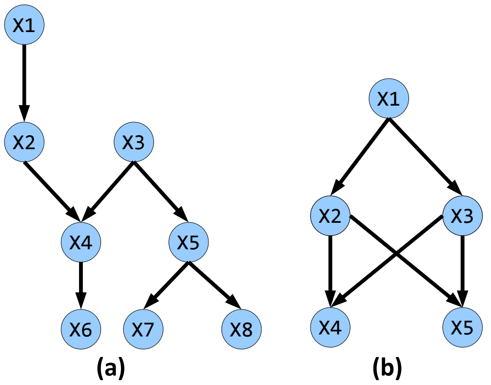


Figure 2.2: (a) A singly connected DAG, (b) A multiply connected DAG

- A **Singly Connected DAG or polytree** is a graph that does not contain any undirected cycles (See Figure 2.2(a)).

- A **Multiply Connected DAG** is a DAG that contains loops (i.e. requires two distinct paths between any pair of vertices in the loop) (See Figure 2.2(b)).

- A **Partially Directed Acyclic Graph (PDAG)** is a graph that contains both directed and undirected edges, with no directed cycles in its directed subgraphs (See Figure 2.3(c)).

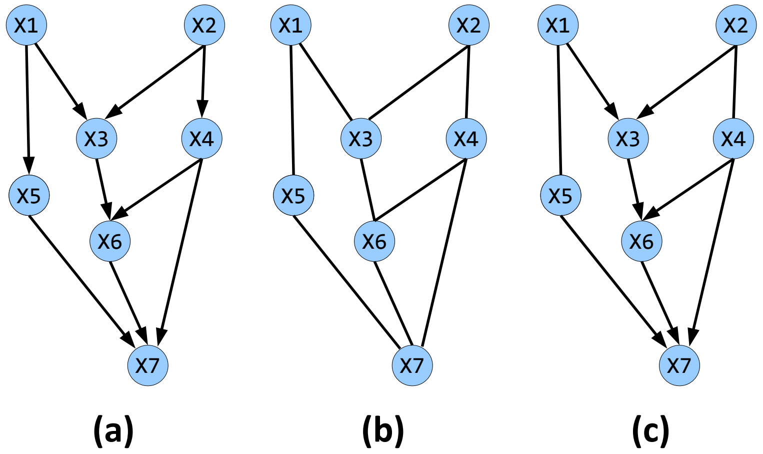


Figure 2.3: (a) example of DAG, (b) the skeleton relative to the DAG and (c) an example of a PDAG.

2.3 Bayesian networks

Over the last decade, Bayesian Networks (BNs) have become a popular representation for encoding uncertain expert knowledge in expert systems [51]. Formally, a BN over a set of variables V consists of two components:

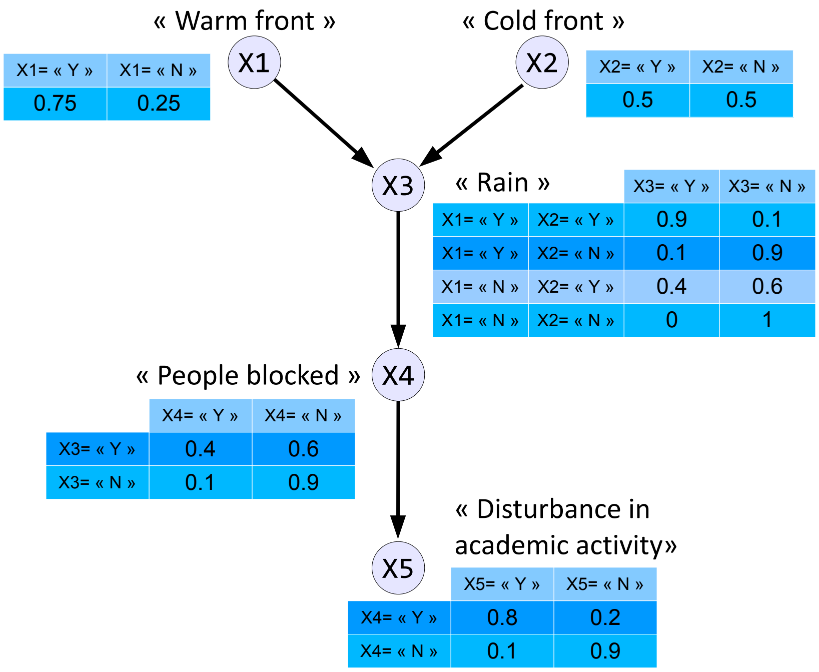
- ***graphical component*** composed of a DAG $\mathcal{G}$ reflecting the dependency relations relative to the modeled domain. BNs encode the conditional independence assumption exposed in Property (2.1).

- ***numerical component*** consisting in a quantification of different links in the DAG by a conditional probability distribution $\mathrm{P}(X_i \mid Pa(X_i))$ of each node X_i in the context of its parents $Pa(X_i)$.

The graphical component corresponds to the structure of the problem, while the numerical component corresponds to the parameters of the model.

Example 2.1. *An illustrative example of a BN in a discrete domain is shown in Figure 2.4.*

It depicts the situation of academic activities through a domain abstracted to five binary variables:



X1= « Y »	X1= « N »
0.75	0.25

X2= « Y »	X2= « N »
0.5	0.5

		X3= « Y »	X3= « N »
X1= « Y »	X2= « Y »	0.9	0.1
X1= « Y »	X2= « N »	0.1	0.9
X1= « N »	X2= « Y »	0.4	0.6
X1= « N »	X2= « N »	0	1

	X4= « Y »	X4= « N »
X3= « Y »	0.4	0.6
X3= « N »	0.1	0.9

	X5= « Y »	X5= « N »
X4= « Y »	0.8	0.2
X4= « N »	0.1	0.9

Figure 2.4: An example of BN modeling the weather and the disturbance in academic activities

- X_1: *represents the event that we have a warm front.*
- X_2: *represents the event that we have a cold front.*
- X_3: *represents the event that it is rainy.*
- X_4: *represents the event that there are people blocked.*
- X_5: *represents whether the academic activities are disturbed.*

For each node in Figure 2.4 is associated a conditional probability table recording the probability of that variable given a particular values combination of its parents. For example, given that we have both cold and warm fronts, the probability that it is rainy is equal to 0.9.

The intuitive interpretation of Figure 2.4 is that X_3 *depends on* X_1 *and* X_2 *however* X_1 *and* X_2 *are independent. Also the two variables* X_1 *and*

X_4 become independent once we know the value of the middle value of X_3. The derived independence statements are essentially due to the application of d-separation rules which will be described in a separate section below.

We will now introduce three basic assumptions that we assume to hold when working with BNs:

- *Causal sufficiency assumption*: This assumption is satisfied if there exist no common unmeasured (also known as hidden or latent) variables in the domain that are influencing one or more observed variables of the domain.

- *Markov assumption*: Each variable X_i in $\mathcal{G}$ is independent of its non-descendants given its parents [98].

$$X_i \perp\!\!\!\perp V \backslash (Desc(X_i) \cup Pa(X_i)) | Pa(X_i). \tag{2.1}$$

  This Markov assumption allows us to obtain a factorized representation of the joint probability distribution (JPD) encoded by a BN via the following chain rule:

$$P(X_1, X_2, ..., X_n) = \prod_{i=1..n} P(X_i \mid Pa(X_i)). \tag{2.2}$$

  Example 2.2. *Given the BN 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, X_4, X_5) = P(X_1) × P(X_2) × P(X_3 | X_1,X_2) × P(X_4 | X_3) × P(X_5 | X_4).
  For instance, P (X_1=Y, X_2=Y, X_3=Y, X_4=Y, X_5=Y) = 0.75 × 0.5 × 0.9 × 0.4 × 0.8 = 0.108

- *Faithfulness assumption*: For a graph $\mathcal{G}$ and a probability distribution P, we say that $\mathcal{G}$ satisfies the faithfulness assumption if, based on the Markov condition, $\mathcal{G}$ entails only conditional independencies in P. The faithfulness assumption allows us to move from probability distribution to graph.

2.3.1 The d-Separation criterion

Let us consider three disjoint sets of variables X,Y and Z, which are represented as nodes in a DAG $\mathcal{G}$. To test whether X is independent of Y given Z in any distribution compatible with $\mathcal{G}$, we need to test whether 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 [88].

In order to define the d-separation criterion, we need first to present the three basic connection structures between variables (see table 2.1).

Hence, the d-separation criterion can be defined as follows [85]:

Definition 2.1. *d-separation:*

A path p is said to be d-separated by a subset of node Z if and only if:

i) p contains serial or diverging connection such that the middle node is in Z , or

ii) p contains a converging connection such that the middle node is not in Z and no descendant of that node is in Z.

The connection between *d*-separation and conditional independence is established through the following theorem [88]:

Theorem 2.1. *If two sets X and Y are d-separated by Z in a DAG $\mathcal{G}$ that satisfied the Markov condition, then X is independent of Y conditional on Z.*

Name	Configuration
serial	$X_i \rightarrow X_j \rightarrow X_k$
diverging	$X_i \leftarrow X_j \rightarrow X_k$
converging	$X_i \rightarrow X_j \leftarrow X_k$

Table 2.1: Elementary structures.

Example 2.3. *Let us consider the DAG represented by figure 2.5*

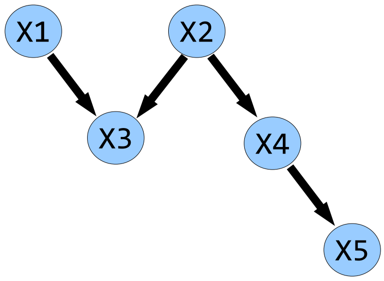


Figure 2.5: Example of DAG (V = X_1, X_2, X_3, X_4, X_5)

- *The nodes X_1 and X_4 are d-separated because the path connecting them contains a converging connection in X_3 and the state of X_3 is unknown. In the other case, X_1 and X_4 will be d-connected given X_3.*
- *However X_2 and X_5 are d-connected 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. We say that X_2 and X_5 are d-separated given X_4.*

2.3.2 The Markov equivalence

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

More formally, Chickering [18] defines the Markov equivalence as follows:

Definition 2.2. *Two DAGs $\mathcal{G}1$ and $\mathcal{G}2$ are equivalent if for every BN1=($\mathcal{G}1$, Θ1), there exists a BN2=($\mathcal{G}2$, Θ2) such that BN1 and BN2 define the same probability distribution, and vice versa.*

Example 2.4. *If we consider the example of figure 2.6, the decomposition of the joint probability distribution for respectively the networks (a) and (b) is as follows:*
$P(X1, X2, X3, X4)_a = P(X1) \times P(X2 \mid X1) \times P(X3 \mid X1) \times P(X4 \mid X2, X3)$.
$P(X1, X2, X3, X4)_b = P(X1 \mid X3) \times P(X3) \times P(X2 \mid X1) \times P(X4 \mid X2, X3)$.

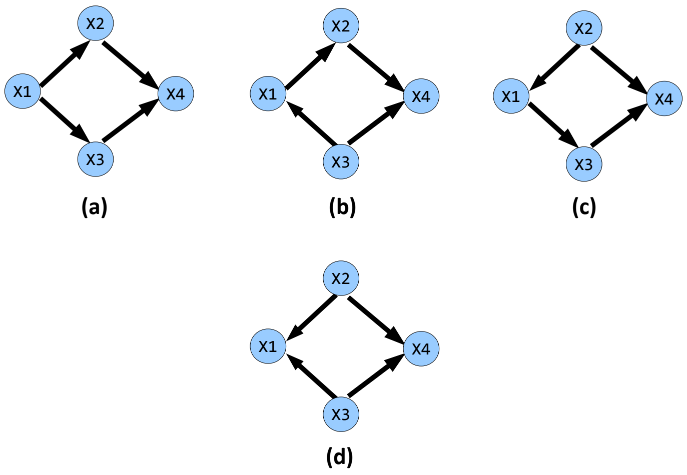


Figure 2.6: Markov equivalence

However,
$P(X1,X2,X3)_b = [P(X3|X1) \times P(X1)/P(X3)] \times P(X3) \times P(X2 \mid X1) \times P(X4 \mid X2, X3) = P(X3|X1) \times P(X1) \times P(X2 \mid X1) \times P(X4 \mid X2, X3) = P(X1, X2, X3, X4)_a$

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 an additional v-structure, is not equivalent to the three others.

Moreover, Verma and Pearl [89] propose the following definition which provides a graphical criterion for determining the equivalence of two DAGs:

Theorem 2.2. *Two DAGs are equivalent if and only if they have the same skeletons and the same v-structures.*

Definition 2.3. *An arc is said to be* reversible *if its reversion leads to a graph which is equivalent to the first one. The equivalence class of DAGs that are Markov equivalent is called CPDAG or essential graph.*

2.3.3 Learning Bayesian networks

One of the most challenging tasks in dealing with BNs is certainly learning their qualitative and quantitative components. The intent of this sub-section is twofold: first of all, we provide a review on principle approaches to learn

BN parameters from data. Secondly, we detail the two main strategies for learning BN structure.

1) **Parameters learning**

Generally, before learning the parameters of a BN, we assumed that the network structure is fixed [1]. So the network parameters can be:

- fixed by a domain expert.
- or estimated from a dataset.

This estimation comes down to estimating the values of all parameters of the conditional distribution $\mathrm{P}(x_i \mid \mathrm{Pa}(x_i))$. We will describe two of the most used approaches in the literature. More details can be found in [80].

Maximum Likelihood Estimation

The *Maximum Likelihood Estimation* (MLE) is the principle of estimating values of parameters that fit the data best. It is one of the most commonly used estimators for fixing the probability of an event $\mathrm{P}(X_i \mid \mathrm{Pa}(X_i))$ using its frequency in the observational data. This gives us:

$$\hat{P}(X_i = x_k | Pa(X_i) = x_j) = \frac{N_{i,j,k}}{\sum\limits_{k} N_{i,j,k}} \tag{2.3}$$

where:
$N_{i,j,k}$ is the number of times that the data contains the event $\{X_i = x_k$ and $Pa(X_i) = x_j\}$.

The set of parameters found by using this method is denoted by $\widehat{\theta}^{MLE}$.

[1] Generally, in most learning algorithms, parameter learning takes place after structure learning. However, since it has less of an emphasis in this dissertation, it is described first. Moreover, most books discuss it first because structure score depends on parameter distribution.

Bayesian estimation

The *Bayesian estimation* consists of finding the most likely parameters $P(X_i|Pa(X_i))$ when assuming that the prior knowledge is expressed by means of a prior joint distribution over the parameters (e.g. *maximum a posteriori* (MAP) estimation). If we assume that each $P(X_i|Pa(X_i))$ follows a multinomial distribution, the conjugate distribution follows a Dirichlet distribution with the parameters $\alpha_{i,j,k}$.

$$\widehat{\theta}_{i,j,k}^{MAP} = \hat{P}(X_i = x_k|Pa(X_i) = x_j) = \frac{N_{i,j,k} + \alpha_{i,j,k} - 1}{\sum\limits_{k}(N_{i,j,k} + \alpha_{i,j,k} - 1)} \tag{2.4}$$

where:
$\alpha_{i,j,k}$ are the parameters of the Dirichlet distribution associated with the prior probability $P(X_i = x_k|Pa(X_i) = x_j)$.

2) **Structure learning**

Many studies [29, 30, 106] have reported that the graphical structure of a network is its most important part, as it reflects the independence and relevance relationships between the concerned variables.

Definition 2.4. *Given a set of variables V and a dataset D_{obs} containing independent and identically distributed instances samples from an unknown distribution P the goal of structure learning is to infer the topology for the belief network $\mathcal{G}$ that is compatible with P.*

This task leads to an NP-hard problem [20], since the number of possible structures (DAGs) to search grows super-exponentially in the number of domain variables. In this context, Robinson [95] derived a recursive function to determine the number of possible DAGs with n variables:

$$f(n) = \begin{cases} 1 & \text{if n=0} \\ \sum_{i=1}^{n}(-1)^{i+1}\mathrm{C}_n^k 2^{i(n-1)} f(n-1) & \text{if n>0} \end{cases} \tag{2.5}$$

Number of variables	Number of possible DAGs
1	1
2	3
3	25
4	543
5	29.281
6	3.781.503
7	1.138.779.265
8	783.702.329.343
9	1.213.442.454.842.881
10	4.175.098.976.430.589.143

Table 2.2: The super-exponential number of DAGs.

In table 2.2, we give an overview of the numbers of possible DAGs with 1 to 10 variables. As this number increases exponentially, it is evident that it will be not feasible, from a computational viewpoint, to exhaustively explore the entire space of DAGs.

That's why heuristic-based methods have been proposed in order to make a trade-off between the structural network complexity and the network accuracy. We distinguish three main approaches for learning BN structure, namely score-based, constraint-based and hybrid ones. All these methods have the limitation that without extra assumptions about the underlying distribution, they can only learn the BN up until its Markov equivalence class.

Constraint-based approach

This first series of structure learning approaches, often called search under constraints, arises from works of different teams: Pearl & Verma for IC and IC* algorithms [88, 89], Spirtes, Glymour & Scheines [98] for the SGS, PC and FCI and recently the BN-PC algorithm of Cheng & all [17].

We will discuss the PC algorithm in details to explain the mechanism of working of such algorithms.

- *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-structures.
- *Edge orientation, edge propagation*: Based on the already oriented edges, apply some orientation rules called Meek rules [76] to infer new arcs until no more edges can be oriented. The PC rules can be summarized as follows [98]:
  R1: *Directing edges without introducing new v-structures:*
  $\forall$ X_i, X_j $\in$ V, if $X_i \longrightarrow X_j$ and X_j and X_k are adjacent, X_i and X_k are not, and there is no arrow into X_j then orient $X_j - X_k$ as $X_j \longrightarrow X_k$.
  R2: *Directing edges without introducing cycles:*
  $\forall$ X_i, X_j $\in$ V, if it exists a directed path between X_i and X_j, and an edge X_i-X_j, then orient it as $X_i \longrightarrow X_j$.
  R3: *Directing edges without introducing cycles:*
  $\forall$ X_i, X_j, X_k, X_l $\in$ V, if $X_k \longrightarrow X_l$ and $X_j \longrightarrow X_l$ and an edge between X_i-X_j, X_i-X_k and X_i-X_l then orient $X_i - X_l$ as $X_i \longrightarrow X_l$.
  R4: *Extended Meek Rule whenever background knowledge is available:*
  $\forall$ X_i, X_j, X_k, X_l $\in$ V, if $X_l \longrightarrow X_k$ and $X_j \longrightarrow X_l$ and an edge between X_i-X_j, X_i-X_k and X_i-X_l then orient $X_i - X_k$ as $X_i \longrightarrow X_k$.
- *CPDAG to DAG.*

In [76], Meek proves that the above rules are proven to be correct and complete subject to any additional background knowledge. However, in most cases, the skeleton algorithm will not produce the correct skeleton and conditioning sets. Therefore, empirical analysis is necessary to understand when and how errors during the skeleton phase impact edge orientation.

Score-based approach

Contrary to the first family of methods which tried to find conditional independencies between variables, the following approaches go either look for the structures which maximize a certain score (i.e. approximation of the marginal likelihood) reflecting the goodness of fit and look for the best structures.

The main limitation with score-based approach lies in the dimension of the space of DAGs, which grows more than exponentially in the number of nodes. This means that an exhaustive search is not feasible in all but the most trivial cases, and has led to an extensive use of heuristic optimization algorithms. Some examples are:

- greedy search algorithms such as hill-climbing with random restarts [21]. These algorithms start from a network structure (usually without any arc) to explore the search space by adding, deleting or reversing one arc at a time until the score can no longer be improved.
- genetic algorithms, which simulate natural evolution through the iterative selection of the "fittest" models and the hybridization of their characteristics [65]. In this case the search space is explored through the crossover (which combines the structure of two networks) and mutation (which introduces random alterations).
- the simulated annealing algorithm, which performs a stochastic local search by accepting both changes that increase or decrease the network score.

Hybrid approach

Hybrid algorithms aim to combine the strengths of both constraint-based and score-based algorithms [24]. The two best-known versions of this family are the Sparse Candidate algorithm (SC) [42] and the Max-Min Hill-Climbing algorithm (MMHC) [105]. Both of these algorithms are based on two principle steps:

- restrict: runs some form of constraint identification algorithm to restrict the search space of graphical solutions for the next phase.
- maximize: seeks the network that maximizes a given score function among the ones that satisfy the constraints imposed by the restrict phase.

2.4 Causal Bayesian networks

The biggest problem when learning BNs from observational data is that we simply do not observe causal relationships. What we really observe is the cause, the effect and the fact that they occur in a fixed pattern. This correlation implies an unresolved causal structure. In order to provide a causal interpretation for BNs, a causal extension appears, with specific properties and assumptions [88, 103].

2.4.1 Definitions and properties

Definition 2.5. ***(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 modeling 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. This makes them more adapted to make causal inference.

A Causal Bayesian Network is defined as a Bayesian network that respects the following central properties:

- Causal Markov condition
- Causal Sufficiency
- Causal Faithfulness

Nevertheless, the discovery of the causal mechanisms that underlie many real world domains is not purely observational and need experimental confirmation.

2.4.2 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.

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.

2.4.3 Kinds of interventions

Different types of interventions [2] have been proposed in the causal framework [88, 98, 107]. They differ depending on how they can be applied and what

[2] Throughout this text the terms intervention and experiment are used interchangeably.

can be learned from the system they are applied to. To be distinguished from normal causal variables, interventions must be exogenous.

Definition 2.6. ***(Exogenous)*** *A variable is exogenous if it is caused by factors or agents from outside the system. More formally, given a set of variables V, X is called exogenous where $X \notin V$ and there does not exist a variable $Y \in V$ such that Y is a cause of X.*

Definition 2.7. ***(Intervention)*** *Given a set of observed variables V, an intervention I on a subset $S \subseteq V$ must satisfy the following conditions:*

- *$I \notin V$ is a variable with two possible states (on/off)* [3] *representing where the intervention can be active or inactive.*
- *I directly manipulates each variable $X \in S$,*
- *I is exogenous to V*
- *When I=off, the joint distribution over V obtains, i.e.*

$$P(V \mid I = off) = \prod_{V_i \in V} P(V_i \mid pa(V_i)) = P(S \mid pa(S)) \prod_{V_i \in V \setminus S} P(V_i \mid pa(V_i)) \tag{2.6}$$

- *When I=on, the conditional distribution over S is manipulated , i.e.*

$$P(V \mid I = on) = P(S \mid pa(S), I = on) \prod_{V_i \in V \setminus S} P(V_i \mid pa(V_i)) \tag{2.7}$$

where

$$P(S \mid pa(S), I = on) = \prod_{X \in S} P^*(X \mid pa(X))$$

and for each $X \in S$, we have:

$$P^*(X \mid pa(X)) \neq P(X \mid pa(X), I = off) \tag{2.8}$$

In CBNs, an intervention variable is represented as an additional variable with direct arrows into each variable in S. There are two types of interventions that can be made: structural and parametric interventions.

[3] The number of possible states taken by the intervened variable may be increased when we have to perform different forms of interventions

The first one is represented as an exogenous variable I_S (a variable without incoming edges) with two possible values (on/off) and a single arrow into the manipulated variable.

Definition 2.8. *Given a set of observed variables V, a structural intervention I_S on a subset $S \subseteq V$ must satisfy the following additional constraints:*

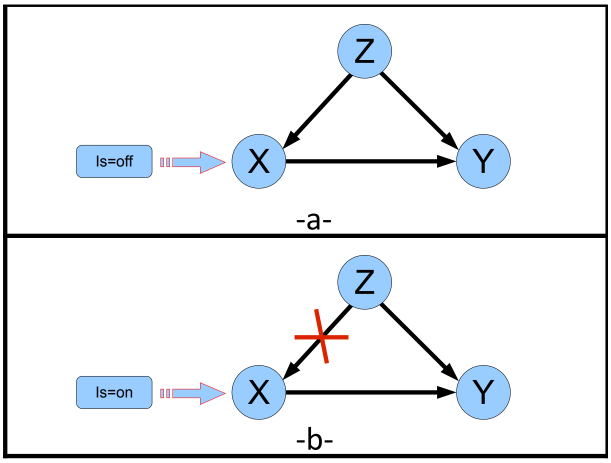


Figure 2.7: Structural experiment

- *When I_S is set to off, we keep the passive observational distribution over the variables.*

- *When we switch to on, all other incoming edges on the intervened variable are removed, and the probability distribution over the manipulated variable will be a determinate function of the intervention.*

The structural interventions aim to make the manipulated variable independent of its normal causes. Various designations are used in the literature to refer this particular type of intervention: randomization [37], surgical interventions [87], ideal interventions [98] or independent interventions [62]. An intervention is called "structural" when it alone completely determines the probability distribution of its targets. The use of structural interventions implies possible changes in the causal structure of the system. The manipulated causal structure is referred to as the post-manipulation graph.

Given a graph G and a set S of variables subject to a structural intervention, the post-manipulation graph is the graph where all the edges incident on any intervened variable (X ∈ S) are removed (See Figure 2.7). This change in causal structure implies a change in the joint probability distribution over the variables [98].

Theorem 2.3. *Let $G = \{V, E\}$ be a DAG and let I be the set of variables in V that are subject to a structural intervention. Then $G_{\overline{man}}$ is the unmanipulated graph corresponding to the unmanipulated distribution $P_{\overline{man}}(V)$ and G_{man} is the manipulated graph, in which for each variable $X \in I$ the edges incident on X are removed and an intervention variable $I_S(X) \rightarrow X$ is added. A variable $X \in V$ is in man(I) if it is subject to an intervention, i.e. if it is a direct child of an intervention variable $I_S(X)$. Then*

$$P_{\overline{man}(I)}(V) = \prod_{X \in V} P_{\overline{man}(I)}(X \mid pa(G_{\overline{man}}, X)) \tag{2.9}$$

$$\begin{aligned} P_{man(I)}(V) = & \prod_{X \in man(I)} P_{man(I)}(X \mid I_{s(X)} = on). \\ & \prod_{X \in V \setminus man(I)} P_{\overline{man}(I)}(X \mid pa(G_{\overline{man}}, X)) \end{aligned} \tag{2.10}$$

Structural interventions are not the only possible type of system manipulation. A weaker form of intervention when it is not necessary to disconnect the experimented variable from its causes can be adopted (See Figure 2.8). This soft version of interventions is also referred to as a partial, conditional or parametric intervention. For coherent notation we will use parametric intervention for designating this type of manipulation.

Definition 2.9. *Given a set of observed variables V, a parametric intervention I_p on a subset $S \subseteq V$ must satisfy the following constraint:*

- *When I_p is set to on, I_p does not make the variable in S independent of their causes in V (it does not break any edges that are incident on variables in S). In the factored joint distribution P(V), the term P(S | pa(S))is replaced by the term P*(S | pa(X), I_p=on), where P*(S | pa(X), I_p=on) ≠ P*(S | pa(X), I_p=off).*

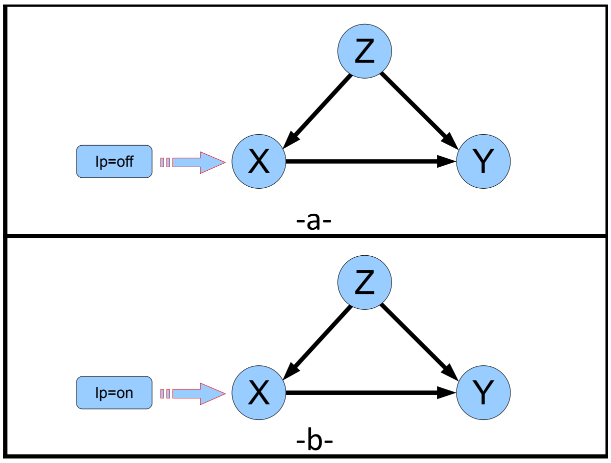


Figure 2.8: Parametric experiment

Otherwise all terms remain unchanged.

Theorem 2.4. *Let $G = \{V, E\}$ be a DAG and let I be the set of variables in V that are subject to a parametric intervention. Then $G_{\overline{man}}$ is the unmanipulated graph corresponding to the unmanipulated distribution $P_{\overline{man}}(V)$ and G_{man} is the manipulated graph, in which for each variable $X \in I$ an intervention variable $I_{p(X)}$ is added with $I_{p(X)} \to X$. A variable $X \in V$ is in man(I) if it is subject to an intervention, i.e. if it is a direct child of an intervention variable $I_{p(X)}$. Then*

$$P_{\overline{man}}(V) = \prod_{X \in V} P_{\overline{man}(I)}(X \mid pa(G_{\overline{man}}, X)) \tag{2.11}$$

$$\begin{aligned} P_{man}(V) = \prod_{X \in I} P_{man}(X \mid pa(G_{\overline{man}}, X), I_{p(X)} = on). \\ \prod_{X \in V \setminus I} P_{\overline{man}(I)}(X \mid pa(G_{\overline{man}}, X)) \end{aligned} \tag{2.12}$$

[33] showed that for N causally sufficient variables N-1 experiments are sufficient and in the worst case necessary to discover the causal structure among a causally sufficient set of N variables if at most one variable can be subjected to a structural intervention per experiment assuming faithfulness. If multiple variables can be randomized simultaneously and independently

in one experiment, this bound can be reduced to log(N) + 1 experiments [34].

2.4.4 The "do" operator

In Pearl framework [88], the notification of external intervention is expressed by the do operator.

Definition 2.10. *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 the seeing and the doing in causal analysis is expressed as follow:

- Conditional probability that variable Y = y when we see that X = x is noted:
  P(Y = y | see(X = x)) = P(Y = y | X = x) = P(y | x)

- Conditional probability that variable Y =y when we set X to x is noted:
  P(Y = y | do(X = x)) = P(y | do(x))

  Two alternatives can be applied: either Y is the direct cause of X and P(y | do(x)) is equal to P(y) (resp. P(x | do(y)) is equal to P(x | y), or the opposite case where we maintain P(y | do(x)) is equal to P(y | x) (resp. P(x | do(y)) is equal to P(x) as Y is the direct effect of intervening on X.

In general, the applicability of the causal inference can be decided using Pearl's do-calculus. This allows finding answers to questions about the mechanisms by which variables come to take on values, or predicting the value of a variable after some other variable has been manipulated. By ensuring that, causal inference could have a major impact on the conclusions we draw in various fields, from health sciences to policy studies passing through AI research.

2.4.5 Conditioning vs. manipulating

The formal distinction between the two notions is an important prelude to the rest of this thesis:

- **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.

To illustrate these two notions, let us consider the following example [98].

Example 2.5. *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. Let's 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 someone 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 probability 0).

In contrast to conditioning, a manipulated probability distribution is not usually a distribution in a subprobability 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, this intervention will not make any change since the Light have not a direct effect on the Switch. So we obtain:

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

Hence, P(Switch=off | do(Light=off)) $\neq$ 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, then 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 did 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 that 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. Thus, discovering the causal relations between variables is a necessary step to correctly inferring the results of manipulations.

2.5 Learning CBNs

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

2.5.1 Active learning for CBN structure

Learning CBNs has recently been incorporated with active learning. There are two formal frameworks covering active learning for CBN structure, namely, Tong and Koller approach [104] and the utility approach developed by Murphy [79].

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 distribution, Tong and Koller [104] consider an active learner that is allowed to conduct experiments. 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 [38].

Murphy [79] proposed a similar technique where different approximations are used to overcome working in the space of DAGs. [79] used MCMC to approximate the belief state $P(G|D^i)$ and importance sampling to calculate the expected utility.

2.5.2 Causal discovery as a Game

[32, 31] presents a theoretic approach in which the causal discovery is considered as a two person game between Nature and the Scientist. The scientist attempts to discover the true causal structure and Nature tries to make discovery as difficult as possible (in term of number of experiments). This approach provides a very general framework for the assessment of different

search procedures and a principled way of modeling the effect of choices between different experiments.

2.5.3 Active learning of causal networks with intervention experiments and optimal design

Geng & He [50] developed a framework for active learning of causal structures via intervention experiments. They discussed two kinds of external intervention experiments: the randomized experiment and the quasi-experiment. In order to reduce the complexity of the causal discovery task, the authors proceed by splitting the Markov equivalence class into subclasses and making experimentations on chain components.

They also proposed two optimal designs of batch (incremental) and sequential interventions. For the optimal batch design, a smallest set of variables to be manipulated have to be found before interventions.The principle drawback of this strategy is that it does not use orientation results obtained by manipulating the previous variables during the intervention process. This weakness will be remedied in the optimal sequential design when the variables are manipulated sequentially such that the Markov equivalence class can be reduced to a subclass with potential causal DAGs as little as possible. They discussed two criteria for optimal sequential designs, the minimax and the maximum entropy criteria.

2.5.4 Learning CBN from mixture of observational and experimental data

Cooper and Yoo [23] proposed another score-based method which can learn the structure from an arbitrary mixture of imperfect observational and experimental data. 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 [22, 52] for observational data alone.

2.5.5 Decision theoretic approach for learning CBNs

Two major approaches can be distinguished:

- *MYCADO approach:* Meganck & al. [77] proposed a greedy approach for learning CBNs from perfect observational data and experiments known as MYCADO (My Causal Discovery) algorithm. This algorithm first assumes as input a perfect observational dataset that can be modeled by a CBN.

  Using traditional structure learning techniques it learns CPDAG from observational data. Then it selects the best experiments to perform in order to discover the directions of the remaining edges. The general overview of MYCADO is given in Figure 2.9.

  The choice of best experiment depends on calculating a utility function U(A_{X_i}), where A_{X_i} (resp. M_{X_i}) denotes performing an experiment on X_i (resp. measuring the neighboring variables).

  The general formula of U(A_{X_i}) is expressed by:

$$U(A_{X_i}) = \frac{Card(Ne_U(X_i)) + Card(inferred(inst(A_{X_i})))}{\alpha cost(A_{X_i}) + \beta cost(M_{X_i})} \tag{2.13}$$

  where measures of importance α and $\beta \in$ [0,1] .

  The number of undirected edges (i.e. $Card(Ne_U(X_i))$) and those susceptible to be inferred in appropriate instantiation among all instantiations of $X_i - Ne_U(X_i)$ (i.e. $Card(inferred(inst(A_{X_i})))$) represent the gained information in the utility function. Clearly, the utility result will be proportional to the experimentation gain and inversely proportional to the cost of performing an action ($cost(A_{X_i})$) and measuring neighboring variables ($cost(M_{X_i})$).

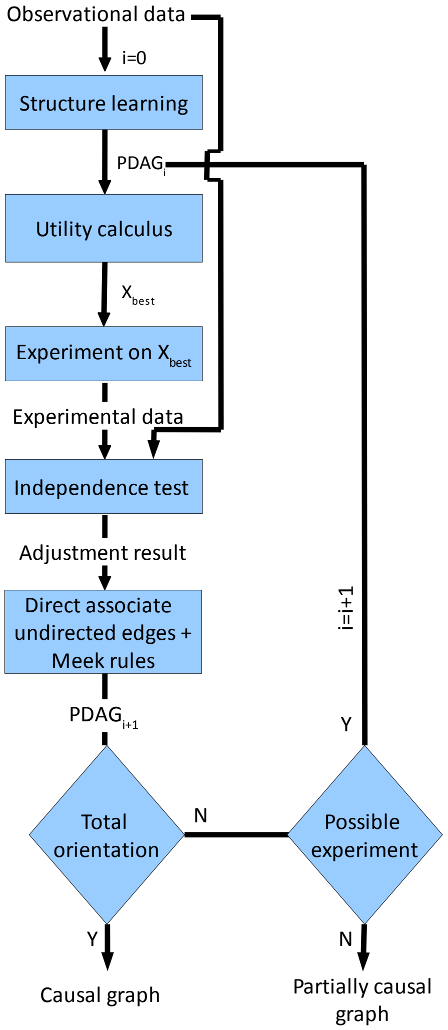


Figure 2.9: MYCADO Algorithm

Hence, depending on the number of undirected neighbors and edges susceptible to be directed by applying the Meek rules, three decision criteria were proposed in [77]:

- 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.

$$X_{best} = \underset{inst(X_i)}{argmax}(Max\ U()) \tag{2.14}$$

- 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.

$$X_{best} = \underset{inst(X_i)}{argmax}(Min\ U()) \tag{2.15}$$

- 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.

$$X_{best} = \underset{inst(X_i)}{argmax}(Exp\ U()) \tag{2.16}$$

- *Learning CBNs from incomplete observational data and interventions :* The basic idea of Borchani et al. approach [10] is to extend the GES-EM [9] algorithm via performing an additional phase in order to discover causal relationships.

  - adaptive approach : where interventions are performed sequentially and where the impact of each intervention is considered before starting the next one. 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 $Ne_U(X_i)$.

  - non-adaptive approach : where interventions are executed simultaneously.

2.5.6 Applications of Causal Discovery with CBNs

When owing the ability to disentangle causality, the BNs can be used in many different domains:

- Explaining human causal reasoning requires supplementing the actual methods developed in computer science with causal domain knowledge reflecting the human behavior [96, 91, 46].
- In the domain of medicine, the identification of the causal factors of diseases and their outcomes, allows better management, prevention and improvement of health care [75].
- More recently, with the advent of the DNA microarrays technology, causal discovery techniques based on microarray data [41] have been proposed in order to build causal networks representing the potential dependencies between the regulations of the genes.
- Engineers use these models and their diagnostic capabilities to detect the cause of defect as early as possible to save cost and reduce the duration of service breach [63].
- Scientists also need CBNs for the domain of ecological prediction and policy analysis [11].

2.6 Conclusion

In this chapter we showed how probabilistic BNs can be extended to represent causal relationships between system variables. Furthermore we showed how this cause to effect interpretation allows causal inference in CBNs.

This edge causal interpretation has the 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 another type of knowledge representation based on semantical modeling.

The process of building or engineering ontologies for use in information systems remains an arcane art form, which must become a rigorous engineering discipline.
Guarino (2002), Evaluating Ontological Decisions With Ontoclean

Chapter 3

Ontology: State of the art

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, additional knowledge can improve the causal discovery.

In many cases, available ontologies provide high level knowledge for the same domain under study [43]. 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 [59], medicine [44], electronic commerce [70] and bioinformatics [2].

Therefore, the semantical knowledge contained in the ontology can turn out of a big utility to improve causal discovery. Reciprocally, the causal knowledge base construction will enable us to relate causal discoveries to ontologies and participate to the ontology evolution.

In section 3.2, we formally introduce ontologies. Section 3.3 discusses how such representation can be semantically enriched from other knowledge sources. Finally, section 3.4 investigates some ways to link ontologies and

BNs.

3.2 Basics on ontologies

A Knowledge-based system (KBs) provides a consistent reasoning framework dotted with an inference engine that deductively reason over a logical language. Ontology is one such kind of semantic driven knowledge based system. There are different definitions in the literature of what should be ontology. The most accepted one was given by [47], stipulating that an ontology is an *explicit* specification of a *conceptualization*. 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 explicitly defined.

Definition 3.1. *In this way, ontology will be defined by:*

- *a set of concepts or classes* $\mathcal{C}$=$\{C_1, ..., C_n\}$ *structured by means of taxonomic (is-a) and partonomic (part-of) hierarchy* $\mathcal{H}$*,*
- *concept properties or attributes,*
- *semantic relations between concepts (*$\mathcal{R}_c$*:* $C_i \times C_j$*),*
- *a set of concept (resp. relation) instances* $\mathcal{I}$ *(i.e. occurrences of classes and semantic relations),*
- *a set of formal axioms* $\mathcal{A}$=$< c_{ik}, c_{jm}, v_n >$ *with* $c_{ik}, c_{jm} \in \mathcal{I}$ *and* $v_n \in \mathcal{V}$ *(i.e. a set of constraints like must, must not, should, should not, etc).*

The first four components are shown schematically in Figure 3.1, where concepts are tagged by yellow circles and instances are marked with blue rhombus. The is-a relations concern inter-related concepts and the non-labeled ones indicate instantiation relationships. We distinguish between two types of causal relations in the ontology. The first ones which are indicated in solid lines build causal connections between the ontology concepts. The other types in dashed lines consider more specific causal relations that exist between concept instances. We restrict the use of semantic relations to only causal ones between concepts since they are the main relations recovered in

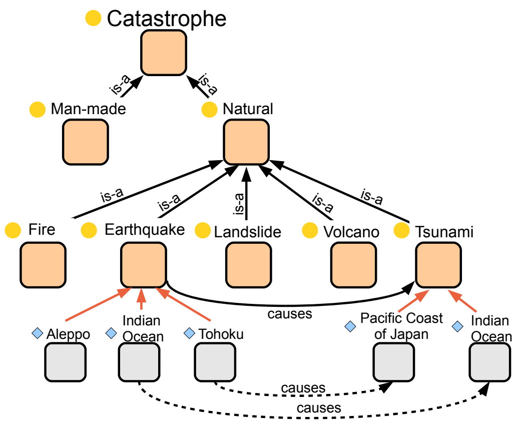


Figure 3.1: An illustrative example of Risk & Catastrophe Ontology

our approach. For more details on the data model and syntax of the OWL ontology language, please refer to Appendix A.

3.2.1 Ontology categories

Ontologies may exist at multiple levels of abstraction. Specifically we distinguish amongst three categories of upper, mid-level and domain ontologies (as illustrated in Figure 3.2).

- An *upper (or foundation) ontology*, is a top-level, domain-independent ontology, from which more domain-specific ontologies may be derived. The concepts expressed in such ontology are intended to be meta, generic and abstract to ensure expressivity for a wide area of domains.

- A *Mid-Level ontology* is designed specifically to serve as the interface between top-level concepts defined in the upper ontology and low-level concepts specified in a domain ontology. In other terms, a mid-level ontology is an upper ontology for a specific domain.

- However, a *domain (or domain-specific) ontology* models a specific field of knowledge, or part of the world. It represents the particular meanings of terms as they apply to that domain. Domain ontologies may also extend concepts defined in both mid-level and upper ontologies.

For tasks that use specific fields of knowledge, it will be more adequate to use domain ontologies instead of upper or mid-level ontologies.

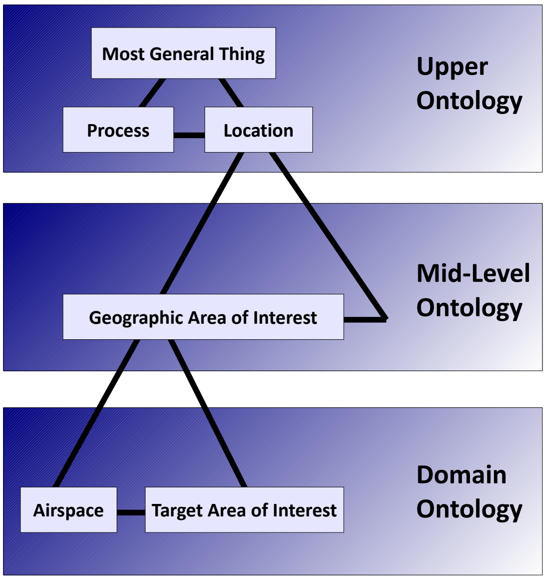


Figure 3.2: The Ontology Categories

3.2.2 Uses of Ontologies

In this section, we provide the basic motivations for using and developing ontologies. For tasks that use a specific field of knowledge, it can be more adequate to utilize domain ontologies instead of more general ones. In doing so, we sub-divide the space of uses for ontologies into the following three categories:

- *Inter-operability*: Many applications based on ontologies address the issues of interoperability in which different information management systems are deployed and different system-users need to exchange data using various software tools. The major contribution for the recourse to ontologies in domains such as enterprise modeling and multiagent architectures is the creation of a shared understanding of common domains allowing applications to agree on the terms that they are using when communicating. Hence, ontologies, if shared among the interoperating applications, allow a semantical data exchange between these tools.

- *System engineering*:

  A shared and consistent understanding of the problems and the tasks at hand can assist in the specification of software engineering project. In this way, software engineering ontologies are developed in order to represent and communicate over software engineering knowledge.

  The development of such "software engineering domain ontology" will allow us to:

  - share and reuse all knowledge accumulated until now in the Software Engineering field;
  - open new avenues to automatic interpretation of this knowledge.

  For example, the SWEBOK project [1](Software Engineering Body of Knowledge [12]), is the result of great effort of declarative and procedural knowledge mining, acquisition and structuring of very diverse documents (scientific papers, congress proceedings, books, chapters, technical reports, technical standards), and of background knowledge from field experts, consultants and researchers. The SWEBOK project team established the project with five objectives: 1) characterize the contents of the software engineering discipline; 2) provide topical access to the software engineering body of knowledge; 3) promote a consistent view of software engineering worldwide. 4) clarify the place and set the boundaries of software engineering with respect to other disciplines such as computer science, project management, computer engineering, and mathematics; 5) provide a foundation for curriculum development and individual certification material.

- *Communication*: Recall that ontologies reduce considerably terminological confusion by providing a unifying framework within an organization. In this way, ontologies enable shared understanding and com-

[1] http://www.computer.org/portal/web/swebok

munication between people with different needs and viewpoints arising from their particular contexts.

3.2.3 Semantic measures on ontologies

Recently, several works highlighted the importance of evaluating taxonomic measures 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* [92], *Sussna's distance* [101] and *Jiang and Conrath's distance* [57]. For the semantic similarity, we find *Leacock and Chodorow's similarity* [69], *Wu and Palmer's similarity* [108] and *Lin similarity* [72], while for semantic relatedness, the most used one is *Hirst and St Onge's relatedness.* See [8] 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* [92]. 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:

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

where:

- $pths(c_i,\, c_j)$: set of paths between the concepts c_i and c_j.
- $len_e(p)$: length in number of edges of the path p.

3.3 Ontology evolution

One critical point in applying ontologies to real-world problems is that domains are changing fast (new concepts, concepts changing their meaning, new relations, new axioms, etc.) and user needs are changing too. Hence, the corresponding ontologies have to evolve as well.

Ontology evolution is the timely adaptation of the ontology in response to a certain need. Several reasons for changing ontology have been identified in the literature. We can summarize them as follows:

- A dynamic change in the modeled domain [99].
- Some need to change the perspective under which the domain is viewed [81]. For example, consider an ontology describing traffic connections, which includes such concepts as roads, cycle tracks, canals, bridges, and so on. If we adapt the ontology to describe not only the bicycle perspective but also a water-transport perspective, the conceptualization of a bridge changes from a remedy for crossing a canal to a time consuming obstacle.
- Discovering a design flaw or change in the focus of the original conceptualization [90].
- Need to incorporate additional functionalities according to changes in the user's need [48].

- New information, previously unknown, classified or otherwise unavailable may become available or different features of the domain may become important [53].

The process of evolution takes ontology from one consistent state to another [16] and can be of two types:

1) *Ontology Population* : When we get new instances of concept(s) already present in the ontology. Only the new instance(s) are added and the ontology is populated.

2) *Ontology Enrichment* : Which consists in updating (adding or modifying) concepts, properties and relations in a given ontology.

Most common changes [16] can be summarized as follows:

- Adding new concepts: This is the most common change in any ontology. New concepts emerge and have to be accommodated in the already existing concept hierarchy.
- Modifying concept hierarchy: In this case the concept in focus might have different hierarchical position to the existing one.
- Changing concept properties: When the concept in focus is already present in the ontology but its properties are different from the existing one.
- Changing concept restrictions: In this case, the concept in focus having restrictions that are dissimilar from those associated with existing concepts.
- Adding new relations (taxonomic or non-taxonomic) between existing concepts.

Example 3.1. *Figure 3.3 shows an example of ontology evolution. Here the modifications to perform upon the ontology concern relation's addition and concept's addition. Hence, the first change requirement occurs in the second ontology level while deleting the is-a relation between the two concepts F2*

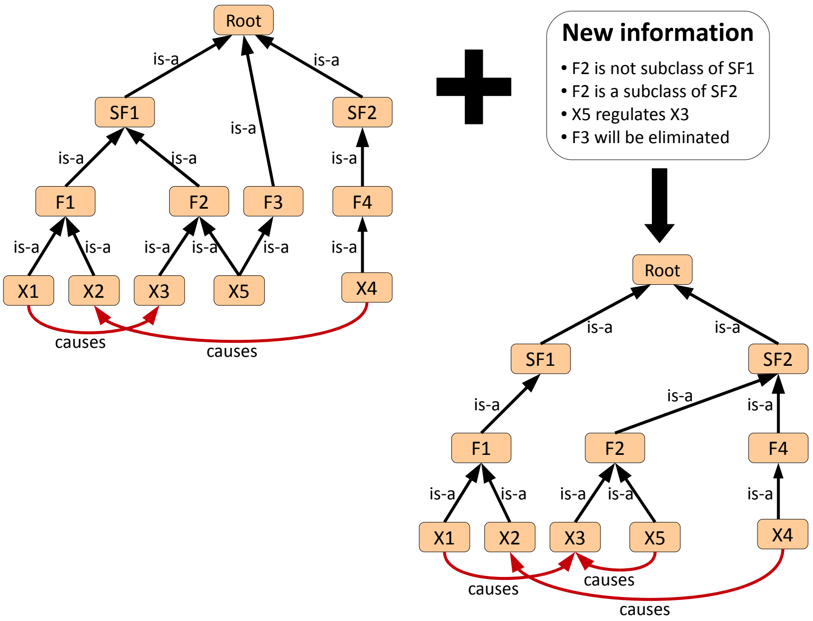


Figure 3.3: Gene Ontology Evolution

and SF1 and replacing it by a new one relating F2 to SF2. Secondly we have to introduce a new causal relation between the concepts X3 and X5. The last modification concerns the deletion of the concept F3 which implies the deletion of the two is-a links relating it to the root concept and the gene X5.

Six phases of ontology evolution have been identified in [99], occurring in a cyclic loop (See figure 3.4). Initially, we have the change capturing phase, where the changes to be performed are determined.

Three types of change capturing have been distinguished: structure-driven, usage-driven and data-driven [49]; these changes are formally represented during the change representation phase. The third phase is the semantics of change phase, in which the effects of the change(s) to the ontology itself are determined; during this phase, possible problems that might be caused to the ontology by these changes are also identified and resolved. This guaranteed the validity of the ontology at the end of the process.

For example, if a concept is deleted, we need to determine how to pro-

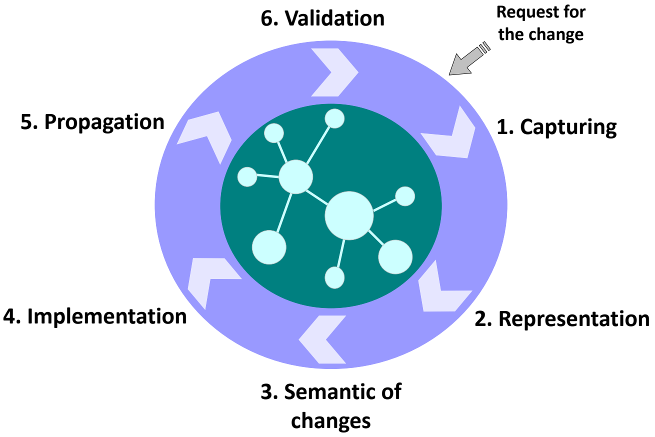


Figure 3.4: Ontology Evolution Process [74]

ceed with its instances (e.g. delete them or re-classify them). In [74], it is suggested that the final decision should be made indirectly by the ontology engineer, through the selection of certain pre-determined evolution strategies, indicating the appropriate action in each case. Other manual and semi-automatic approaches are also possible [49].

The change implementation phase follows, where the changes are physically applied to the ontology, the ontology engineer is informed of the changes and the performed modifications are logged using appropriate tools guaranteeing atomicity, consistency, isolation and durability of changes [49]. When this step is achieved, all these changes need to be propagated to all dependent elements; this is the role of the change propagation phase. Indeed, when an ontology is changed, all dependent applications may not work correctly. An ontology evolution approach has to recognize which change in the ontology can affect the functionality of those applications and to react correspondingly.

Finally, when reviewing ontology changes, further problems may appear; in this case, we need to start over by applying a new evolution process until reaching the ontology stability.

3.4 Links between ontologies and Bayesian networks

In this section, three topics are discussed to illustrate the possible interactions that can be made between ontologies and BNs. For each topic, we order the associated approaches from the most general to the most specific.

3.4.1 Ontology mapping

The problem of aligning heterogeneous ontologies via semantic mappings has been identified as one of the major challenges of semantic web technologies. In order to enable interoperability among heterogeneous information sources, we often need to establish mappings between ontologies. These mappings capture the semantic correspondence between concepts in ontologies.

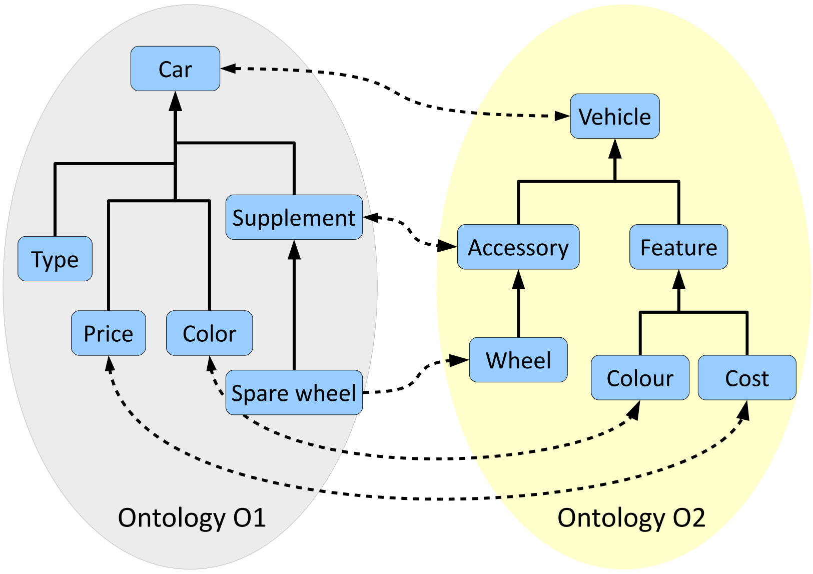


Figure 3.5: Example of two heterogeneous ontologies and their mappings

Figure 3.5 shows an example of metadata heterogeneity between the two ontologies *O1* and *O2*. It is clear that a number of similarities exists among these two ontologies (the dashed line represents a reasonable mapping between similar concepts). For instance, the two concepts Color and Colour will be matched since the word colour is spelled as color in American English. The meaning of the two words is the same. Also, a subsumption link will be added between the two entities Wheel and Spare wheel. And finally, entities {Car, Price, Supplement} will be respectively matched to {Vehicle, Cost, Accessory}.

The instance heterogeneity concerns the different representations of instances. For example, a price can be represented as '30000 dinars' and also as '30000 TND'. We note that many efforts have been placed on the problem of metadata heterogeneity and few works focus on instance heterogeneity.

Most of the existing ontology-based semantic integration approaches try to provide exact mappings in an automatic or semi-automatic way. In this way, many works have tried to increase ontology mapping precision with incorporating uncertainty into the mapping process. Three approaches that use BNs for ontology mapping have been recently reported.

1) RiMOM

[102] formalize the ontology mapping as a decision making problem with the aim to discover the optimal mapping with the minimal risk. To perform this task, an approach called RiMOM (Risk Minimization based Ontology Mapping) were proposed and the problem have been formulated using Bayesian decision theory.

RiMOM treat the ontology mapping as a classification problem and uses for this purpose the Naive bayes technique where the observations (i.e. set of samples) are all entities in the two ontologies to map. Entities $\{e_{i1}\}$ in the first ontology are viewed as samples and entities $\{e_{i2}\}$ in the second one are viewed as classes. So each entity e_{i1} can be classified to one "class" e_{i2}. This also means that entity e_{i1} is mapped onto entity e_{i2}. For recognizing the optimal mapping, they use $p(e_{i2} \mid e_{i1})$ to denote the conditional probability of the entity e_{i1} being mapped onto entity e_{i2}. In this way, they define actions as all possible mappings (i.e. all candidate mappings) in order to find the optimal mapping (i.e. the action with minimal risk).

They also include the two ontologies *O1* and *O2* in the conditional probability $p(e_{i2} \mid e_{i1}, O_1, O_2)$, which means that not only the information relative to entities themselves but also information in O_1 and O_2 will be considered for calculating the mapping risk.

2) OMEN

[78] developed a tool called OMEN (Ontology Mapping ENhancer) which uses Bayesian networks in order to enhance existing ontology mappings by deriving missed matches and invalidating existing false matches. First of all, they have to build a BN with the concept mapping. This BN uses a set of meta-rules based on the semantics of the ontology relations that expresses how each mapping affects other related mappings. Next, the initial probability distribution will be used to infer probability distributions for other mappings.

The following summarizes the OMEN algorithm:

3) Bayes OWL

BayesOWL [28] is one of those probabilistic frameworks which aim to model uncertainty in semantic web. This framework provides a set of translation rules in order to convert OWL ontologies into a DAG of BN. The general principle underlying these rules is that all classes (specified as "Objects" in RDF triples of the OWL file) are translated into nodes in BN, and an arc is drawn between two nodes in BN if the corresponding two classes are related in the OWL file. Information about the uncertainty of the classes and relations in an ontology is represented as conditional probability tables (CPTs) which can be either provided by domain expert or learned from web data, by using text classification programs. With BayesOWL, concept mapping can be processed as some form of

Algorithm 1 OMEN algorithm

```
Input: source ontologies $O$ and $O'$, initial probability distribution for matches
Steps:
    a) If initial probability of a match is above a given threshold, create a node representing the match and mark it as evidence node.
    b) Create nodes in the BN graph representing each pair of concepts $(C,C')$, such that $C \in O$ and $C' \in O'$ as a node in the graph and the nodes are within a distance $k$ of an evidence node.
    c) Use the meta-rules to generate CPTs for the BN.
    d) Run the BN.
Output: a new set of matches.
```

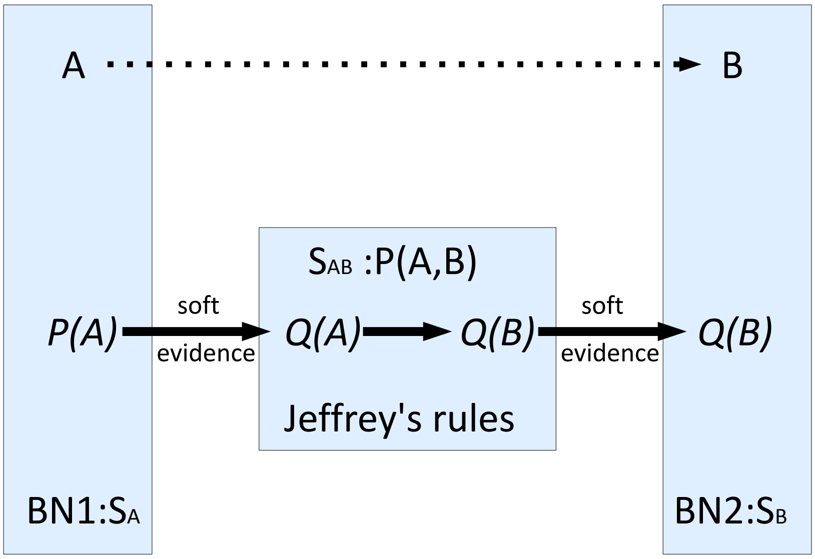


Figure 3.6: BayesOWL: Concept mapping process [28]

probabilistic evidential reasoning between the BN1 and BN2, translated from the Ontologies 1 and 2. This technique allows the two BNs to exchange beliefs via variables that are similar but not identical. First of all, they assume that the similarity information between concepts A from ontology 1 and B from ontology 2 is captured by the joint distribution P(A,B). Three probability spaces will be defined: S_A and S_B for BN1 and BN2, and S_{AB} for P(A,B). The mapping from A to B amounts to determine the distribution of B in S_B, given the distribution P(A) in S_A under the constraint P(A,B) in S_{AB}.

To propagate probabilistic influence across these spaces, they apply the Jeffrey's rule [86] and treat the probability from the source space as soft evidence to the target space. As depicted in Figure 3.8, mapping A to B is accomplished by applying Jeffrey's rule twice, first from S_A to S_{AB}, then S_{AB} to S_B.

3.4.2 Probabilistic Ontologies

Uncertainty is an inevitable feature in most world domains since the available information is mostly incomplete and often imprecise. The Venn diagram of figure 3.7 illustrates some countries' memberships in regional and continental communities. A crisp *partOf* meronymy cannot express that Turkey is to some degree part of all three communities in the diagram (Europe, Asia and Middle East) or traduce the Israeli occupation in both Palestine and Lebanon.

To overcome the difficulty arising from using the crisp logics, an extension of ontologies is required in order to capture uncertainty knowledge about concepts, properties and relations and support reasoning with inaccurate information.

Along this direction, many works in the past have attempted to apply different formalisms such as Fuzzy logics [110], Rough Set theory [83] and Bayesian probability into the ontology definition and reasoning. In this subsection, we will investigate different approaches in the literature that addressed this problem by proposing probabilistic extensions to ontologies.

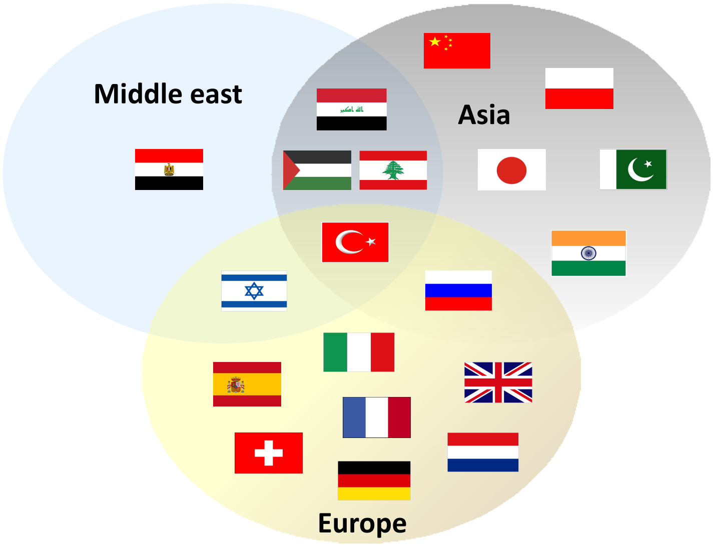


Figure 3.7: A Venn diagram illustrating countries' memberships in regional and continental communities

1) Ding & Peng OWL Probabilistic extension:

   Ding and Peng [27] proposed an approach when they augment the OWL language to allow additional probabilistic markups so that probability values can be attached to individual concepts and properties. For example, if A and B are classes: P(A) is interpreted as the probability that an arbitrary individual belongs to class A. P(A | B) traduces the probability that an individual of class B belongs to A. For this purpose, they define three kinds of OWL classes (owl:Class): "PriorProbObj", "CondProbObjT" and "CondProbObjF".

   They also developed a set of rules to translate a probability-annotated ontology into a BN structure. The general principle underlying these rules is that all classes (specified as "subjects" and "objects" in RDF triples of the OWL file) are translated into nodes in BN, and an arc is drawn between two nodes in BN only if the corresponding two classes are related by a 'predicate' in the OWL file with the direction from the superclass to the subclass if it can be determined. One of the main advantage of this probabilistic-extended ontology is that it can support common ontology-related reasoning tasks as probabilistic inferences.

2) OntoBayes

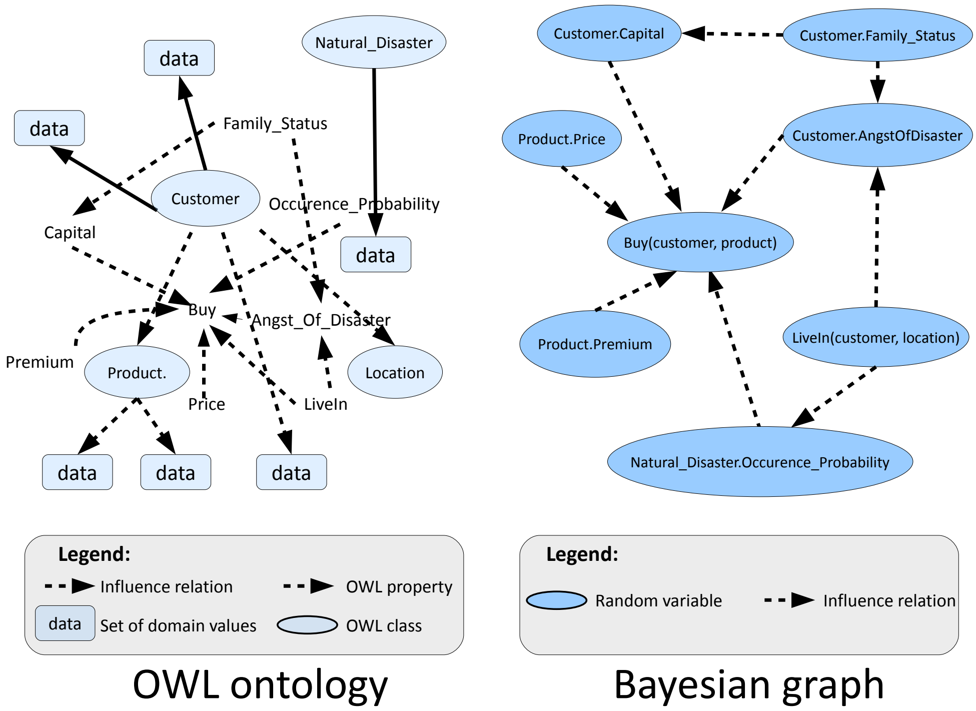


Figure 3.8: Ontobayes: Building a BN from an OWL ontology (insurance ontology)

[109] have proposed the OntoBayes approach, an ontology-driven Bayesian model for uncertain knowledge representation, to extend ontologies to probability-annotated OWL in decision making systems. First of all, they made a probabilistic extension of OWL in order to specify probability-annotated classes or properties. More precisely, they define three OWL classes: 'PriorProb', 'CondProb' and 'FullProbDist'. The first two classes are defined to identify the prior probability and conditional probability respectively. They have a same datatype property 'ProbValue', which can express the probabilistic value between 0 and 1. The last class is used to specify the full disjoint probability distribution. Then they introduce an

additional property element <rdfs:dependsOn> to markup dependency between class properties in an OWL ontology. Hence any expression for BNs in OntoBayes is a collection of triples, each consisting of a subject, a predicate and an object, where the predicate is constantly the primitive <rdfs:dependsOn> and the subject and object are properties. Using this dependency triples, they enable the BN construction by the following rules:

- Extracting all dependency triples from an ontoBayes ontology.
- Merging all triples: all nodes with a same identifier are composed into one single node. For example, if there are two triples A → B and B → C, they can be merged into a BN with only one node B such as A → B → C.

By this way, OntoBayes model preserves the ability to express meaningful knowledge in very large complex domains and extent ontologies to probability-annotated OWL to facilitate meaningful knowledge representation in uncertain systems.

3) PR-OWL 1.0:

The logical basis of PR-OWL 1.0 is MEBN logic [66], which combines Bayesian probability theory with classical First Order Logic. Probabilistic knowledge is expressed as a set of MEBN fragments (MFrags) organized into MEBN Theories. An MFrag is a knowledge structure that represents probabilistic knowledge about a collection of related hypotheses. Hypotheses in an MFrag may be *context* (must be satisfied for the probability definitions to apply), *input* (probabilities are defined in other MFrags) or *resident* (probabilities defined in the MFrag itself). An MFrag can be instantiated to create as many instances of the hypotheses as needed (e.g. an instance of the 'EducationLevel' hypothesis for each person as depicted in Figure 3.9). Instances of different MFrags may be combined to form complex probability models for specific situations. A MEBN theory is a collection of MFrags that satisfies consistency constraints ensuring the existence of a unique joint probability distribution over instances of the hypotheses in its MFrags.

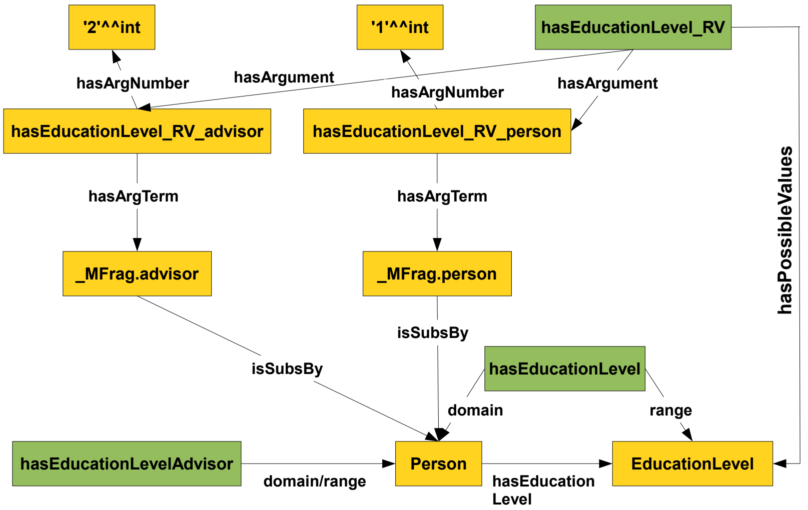


Figure 3.9: Education knowledge domain representation using PR-OWL 1.0

A probabilistic ontology must have at least one individual of class MTheory, which is a label linking a group of MFrags that collectively form a valid MEBN Theory. Individuals of class MFrag are comprised of nodes, which can be resident, input, or context nodes. Each individual of class Node is a random variable and thus has a mutually exclusive and collectively exhaustive set of possible states. In PR-OWL 1.0, the object property hasPossibleValues links each node with its possible states, which are individuals of class Entity. Finally, random variables (represented by the class Nodes in PR-OWL 1.0) have unconditional or conditional probability distributions, which are represented by class Probability Distribution.

4) PR-OWL 2.0:

The major problem with PR-OWL 1.0 is the fail to achieve full compatibility with OWL (See Figure 3.10). Therefore, [15] have recently proposed a new syntax and semantics, defined as PR-OWL 2.0, which improves compatibility between PR-OWL 1.0 and OWL in two impor-

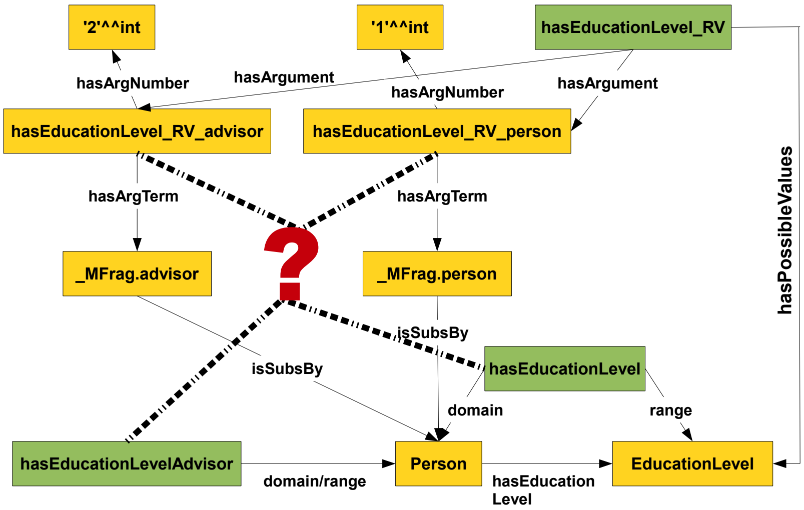


Figure 3.10: PR-OWL 1.0 lack of mapping from arguments to OWL properties.

tant respects. First, PR-OWL 2.0 formalizes the association between random variables from probabilistic theories with the individuals, classes and properties from OWL. Second, PR-OWL 2.0 allows values of random variables to range over OWL datatypes.

5) Holi & Hyvönen approach for computing overlaps:

[54] presents a probabilistic method to represent overlap in taxonomies and to compute the overlap between concepts. Thus an overlap table can be created for every concept in the taxonomy. The authors give, as example, the overlap table of Lapland 3.1 based on the Venn diagram of figure 3.11. The Overlap column lists values expressing the mutual overlap of the selected concept and the other referred concepts, i.e. $Overlap = \frac{|Selected \cap Referred|}{Referred}$. These values will be then used as measure of mutual overlap.

So their method consists of two main parts:

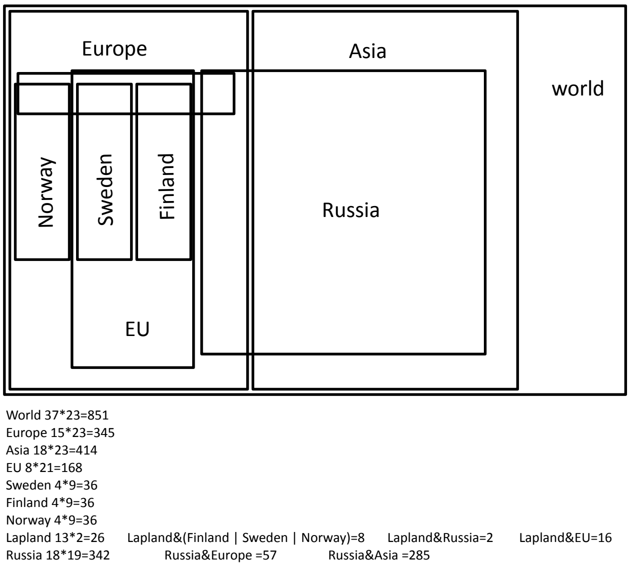


Figure 3.11: A Venn diagram illustrating countries, areas and their overlap [54]

- A graphical notation by which partial subsumption and concepts can be represented in a quantified form.
- A method for calculating degrees of overlap between the concepts of a taxonomy. Overlap is quantified by transforming the taxonomy into a BN, where nodes are classes, arcs are represented by the $rdf : subClassOf$ property and CPTs are fixed using the measures of mutual overlap.

6) MENTOR (Web Adaptive Educational Environment):

[71] presents a study of MENTOR, a web Adaptive Educational Environment (WBES), where the learner's needs and preferences are diagnosed using an ontology-based Bayesian network approach during the learning process (See Figure 3.12).

Firstly, the proposed method uses an OWL ontology to store the Affective Knowledge regarding the learner such as personality, mood and emotions. In this Affective Ontology, we find the Affective_Model class,

Selected	*Referred*	*Overlap*
Lapland	World	26/851=0.0306
	Europe	26/345=0.0754
	Asia	0/414=0.0
	EU	16/168=0.0953
	Norway	8/36=0.2222
	Sweden	8/36=0.2222
	Finland	8/36=0.2222
	Russia	2/342=0.0059

Table 3.1: The overlap table of Lapland according to figure 3.11 [54].

Affective_Tactic class and Emotional_State class. The first class represents the attributes and preferences of the learner. The second represents the affective tactics and the third represents the current emotional state of the learner which can be positive, negative or neutral.

This ontology is extended to deal with uncertainty so that a BN can be constructed from it. To express the affectively uncertain information the OWL classes are defined: '*Pri_Prob*', '*Cond_Prob*' and '*Jnt_Prob*' which identify the prior probability, the conditional probability and the joint probability respectively. The conditional probability distribution for the Affective Tactic given the Learner's Emotional State ET is defined as P(AT | ET).

The transformation into a BN uses a set of rules. They first introduce a property element <owl:Dependent> to specify dependency information in an OWL ontology. All classes of the ontology are converted into nodes in the BN using a set of transformation rules. Such strategy allows them to easily infer the values of the nodes corresponding to the Affective information of the learner's model. This model supplies them with evidences, for selecting the appropriate affective tactic given the values of the Affective model node.

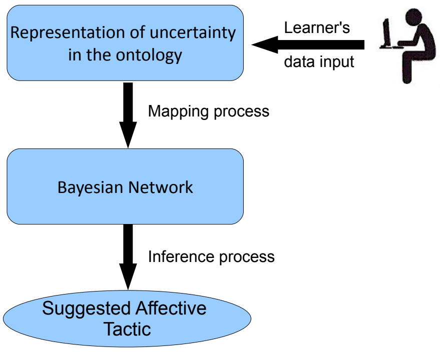

Representation of uncertainty in the ontology
Learner's data input
Mapping process
Bayesian Network
Inference process
Suggested Affective Tactic

Figure 3.12: Mentor Model [71]

3.4.3 BN construction using Ontologies

Ontologies provide a potential knowledge source which could be exploited to facilitate the creation of the BN structure. Recently, there have been some researches to construct a BN (semi-)automatically from ontologies. The challenging tasks encountered in all of these works are:

- The determination of the variables,
- The determination of relationships between variables,
- The calculation of the CPTs for each node.

In this subsection, we give an overview of the principle works which investigate such problem.

1) Constructing BN automatically using ontologies :

   [26] defined a new ontology of BN concepts and link this to the original domain ontology. In order to automatically construct BNs using ontologies, they expose the following correspondences between the two formalisms:

   - Concepts → nodes,
   - Concept attributes → CPTs,
   - Inheritance relations → links.

As shown in Figure 3.13, all concepts of interest for the BN inherit from a node in this new BN ontology. The root concept of the BN ontology is the BNnode. In order to create the BN, an instance of each leaf class which inherits from the BNnode class is created. To describe the generic BNnode, a set of properties and relations have to be defined. The two relations (hasParentNode and hasDelayParentNode) define the influential links between this BNnode instance and other BNinstances. The other properties include name, CPT, state names and levels.

In addition to the basic BNnode concept, the BN ontology may contain additional BN concepts. The domain ontology consists of the two domain concepts subConceptOfNoInterest and subConceptOfInterest and their parent concept Concept1. According to this figure, between the root node and a conceptOfInterest node, there are two intermediate concepts: BehaviourModelNode and Concept1Node. The BehaviourModelNode concept represents the characteristics of BN nodes required for a particular application. The Concept1Node concept defines characteristics of Concept1 instances which should be treated in a particular way. The BN arcs are automatically generated from the domain ontology using a set of inheritance relations and construction rules.

We note that [26] approach does not delve into the area of estimating CPTs. They supposed that BN CPTs are learnt incrementally and online from a live feed of network event data.

2) Ontology-based generation of BNs:

The [36] approach is similar to the previous approach [26]. The main difference between them is that Fenz & al. construct the BN directly from existing domain ontologies and do not require any BN-specific ontology extensions. They used the security ontology which provides detailed knowledge about threat, vulnerability and control dependencies to build up the corresponding BN. Their approach is based on a set of correspondences:

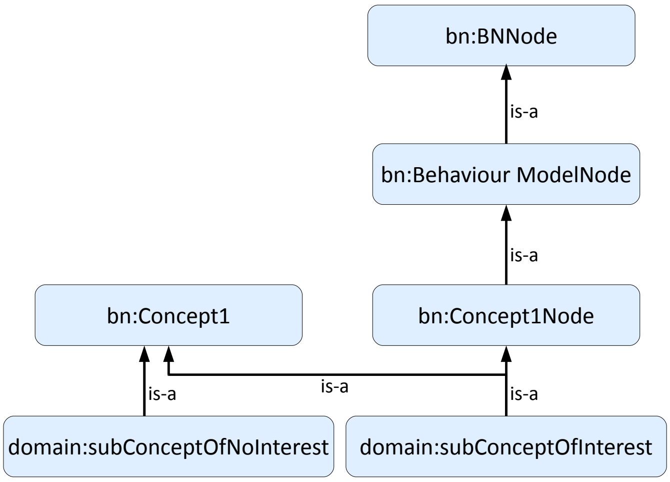


Figure 3.13: Generic Domain Ontology with BN concepts [26]

- Concepts → nodes: The ontology concepts, which are relevant to the considered problem and should be represented in the Bayesian calculation schema are selected to establish the nodes of the BN.
- Ontological relations → links: Ontology relations starting and ending between the selected concepts are used to establish the links between the BN nodes. While the potential relations can be derived automatically from the ontology, the link direction requires the human interpretation of the ontological relation.
- Axioms → node scales and weights: Scale- and weight-relevant axioms are used to determine potential states and weights of the BN nodes.
- Instances → findings: Instances of concepts which are represented by the BN's leaf nodes are used to derive and enter concrete findings in the BN.

The main limitation of this approach is that no strategy regarding the generation of CPTs is given.

3) Ontology-based semi-automatic construction of BN models for diagnosing diseases in e-health applications:

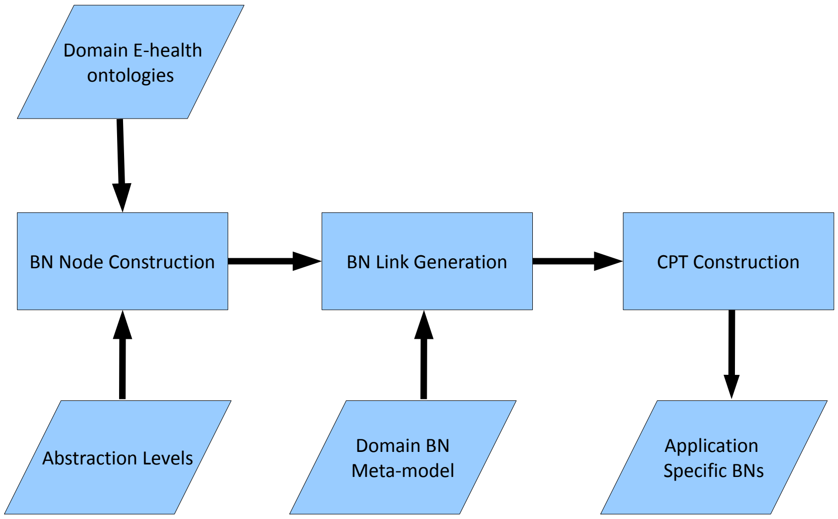


Figure 3.14: The overall processes of the Jeon & Ko approach for BN construction [56]

[56] developed a semi-automatic BN construction system based on e-health ontologies. Their system allows developers to select abstraction levels in e-health ontologies to specify the areas that are mostly useful in diagnosing diseases for an e-health application.

Their approach is still in its early research stage. This is essentially due to the lack of reliable correspondences between the two formalisms. The only two correspondences that are treated are presented as follows:

- Concepts → nodes:

  BN nodes are selectively constructed from the selected ontology areas using a set of rules. For instance, when a class does not have any subclasses, their system constructs BN nodes with the true and false states to represent absence or presence of the situation denoted by the ontology class. However, when the node has subclasses, they generate another BN, a sub-BN, which only contains nodes constructed by the subclasses.

- Causal links between ontologies → links: Using an application-specific meta-model, BN nodes can be linked with each other semi-automatically. According to [56] approach, the domain BN meta-model describes the cause-and-effect relationships between two ontologies in a specific domain and enables the construction of links between the BN nodes. All the construction steps are illustrated in Figure 3.14.

3.5 Some critiques of the former approaches

The majority of the previous approaches that tried to combine BNs and ontologies are still on an early stage of development and research. That is they mainly focus on the theoretical aspects without any intent to test their approaches on real or simulated data. Moreover, they lack the capability of describing potential applications where their approaches would prove valuable and even necessary.

The second limit consists on the use of traditional BNs without regard to any other extensions (Dynamic BNs, Hierarchical BNs, causal Bayesian networks, etc.). Due to this lack of specialization, the correspondences between the two formalisms can be returned to the more general concepts without focusing on specific details. It is worth noticing that in most of the cases, the BN-Ontology cooperation is used to enhance the probabilistic inference. However, they do not make any explicit use of traditional structure learning algorithms.

Finally, we note that the cooperation between BNs and ontologies in all previous contributions is beneficial on only one way (i.e. BN –> ontology or ontology –> BN). One possible direction of research is to develop cyclic strategies which propose a real cooperation in both ways.

3.6 Conclusion

In this chapter, we provide some quick background on ontology basics, uses and evolution. We also present concrete approaches for combining the use of ontologies and BNs. The next chapter will be devoted to presenting our contribution aimed at addressing the same problem.

Serendipity: the making of pleasant discoveries by accident.
The Oxford American Dictionary

Chapter 4

SEMCADO: an iterative causal discovery algorithm for ontology evolution

4.1 Introduction

WE have previously 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. Sometimes, this semantical knowledge can turn out of a big utility to improve causal discovery.

This chapter is devoted to introduce a new algorithm, referred to by SEMCADO (Semantical Causal Discovery), to integrate ontological knowledge for more efficient causal discovery.

This chapter is divided into two major sections: Section 4.2 presents the main principles we suggest to develop our approach. Section 4.3 provides a complete description of the SemCaDo algorithm.

4.2 SEMCADO Principles

This section includes all necessary theoretical foundations for the new method and modalities that enable their translation into a practical algorithm.

4.2.1 Serendipity through design

Generally, in the research field, scientific discoveries represent a payoff for years of conservation works. This affirmation did not exclude the case of other important discoveries that are made while researchers were conducting research in totally unrelated fields.

The examples are abundant from Nobel's flash of inspiration while testing the effect of dynamite, to Pasteur brainstorm when he accidentally discovered the role of attenuated microbes in immunization. In fact, much of our understanding and causal discoveries comes from scientific serendipity (i.e. the manifestation of creativity in which inspiration comes from curiosity and unexpected opportunities).

Scattered over many different areas, there is much literature about how utilizing aspects of serendipity to stimulate creativity. Scientific knowledge [93, 94], web search [1, 13] and information retrieval [111, 35] have all discussed opportunities for insight through coincidences. The search for Serendipity continues with this work.

So instead of treating serendipity as arcane, mysterious and accidental, we surround the ability of computers to optimize the opportunities for insight. The idea here is to investigate some ways to combine the power of CBNs and ontologies, presented in previous chapters. Our main aim is to propose a new causal discovery algorithm to promote and stimulate fortunate discoveries when performing experimentations. To this end two collaborative and complementary strategies are conceivable:

- Build a CBN using ontological knowledge.
- Enrich the ontology by exploiting causal discoveries from the CBN.

In what follows, we assume that:

- Only a single domain ontology should be specified for each causal discovery task.
- The causal graph nodes and their corresponding ontology concepts have the same designations.
- The ontology evolution should be realized without introducing inconsistencies or admitting axiom violations.

The principle of the proposed causal learning algorithm, referred to by SEMCADO (Semantical Causal Discovery), is to benefit of the semantical distance calculus on the corresponding ontology while keeping the same decision criteria used in mycado (see section 2.5.5).

Once the causal discovery step achieved, the results of experimentations can be re-used via an ontology evolution process as shown in figure 3.4. This knowledge acquisition technique provides on the one hand customized domain ontology, and on the other hand an updated ontology which can be used for a variety of semantic tasks such as knowledge management, information retrieval and so on.

4.2.2 CBN-Ontology correspondence

One of the main motivations when realizing this work is to highlight and exploit the similarities between CBNs and ontologies. This is particularly true when comparing the structure of the two models as proposed in Figure 4.1 and Table 4.1:

First, for each CBN node corresponds a single concept from the domain ontology. Accordingly, the correspondence between the two models in term of causal links will be as follows:

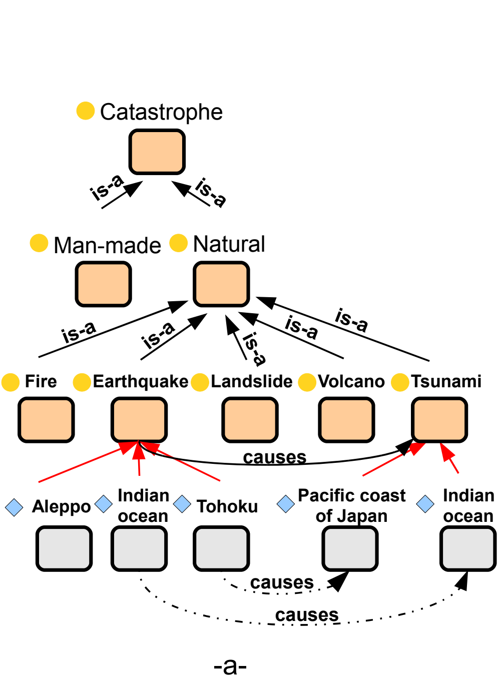


Figure 4.1: CBN-Ontology correspondances

- A causal dependency represented by a directed link in the CBN will be traduced by a specific causal relation between the appropriate concepts in the ontology. In figure 4.1, we show that the domain ontology can provide causal logical links between concepts and instances. In this work, we will only deal with the causal relations between concepts but this did not exclude the possibility to adapt our correspondences according to the context of application.

- Reciprocally, a causal relation between two concepts in the ontology will be traduced by a directed link between the corresponding CBN nodes.

On a more fine-grained level, we can associate both observational and experimental data to the state instantiations of the ontology concepts. All these correspondences lead to a much larger duality between causal inference and logic rule reasoning when using the ontology axioms. Nevertheless, this form of parallelism between the two formalisms may be expressed differently depending on the context of application.

Table 4.1: The main correspondences between causal Bayesian networks and domain Ontologies.

CBN	Ontology
Nodes ($\mathcal{X}$)	Concepts ($\mathcal{C}$)
Causal dependencies ($\mathcal{E}$)	Semantic causal relations ($\mathcal{R}_c$)
Observational & experimental data ($D_{obs/exp}$)	Concept instances ($\mathcal{I}$)
Causal inference	Logic rule reasoning

4.3 SEMCADO Sketch

The general overview of the SEMCADO algorithm is given in Figure 4.2.

So as inputs, SEMCADO needs a perfect observational dataset and a corresponding ontology. Then it will proceed through:

4.3.1 Learning a partially directed structure using traditional structure learning algorithms and semantical prior knowledge

The ontology in input may contain some causal relations in addition to hierarchical and semantic relations. Those causal relations should be integrated from the beginning in the structure learning process in order to reduce the task complexity and better the final output. More precisely, each direct cause to effect relations will be incorporated as constraints when using structure learning algorithms. Our main objective is to narrow the corresponding search space by introducing some restrictions that all elements in this space must satisfy.

In our context, the only constraint that will be defined is edge existence. All these edge constraints can easily be incorporated in usual BN structure learning algorithm [25]. Under some condition of consistency, these existence restrictions shall be fulfilled, in the sense that they are assumed to be true for the CBN representing the domain knowledge, and therefore all PDAGs must necessarily satisfy them.

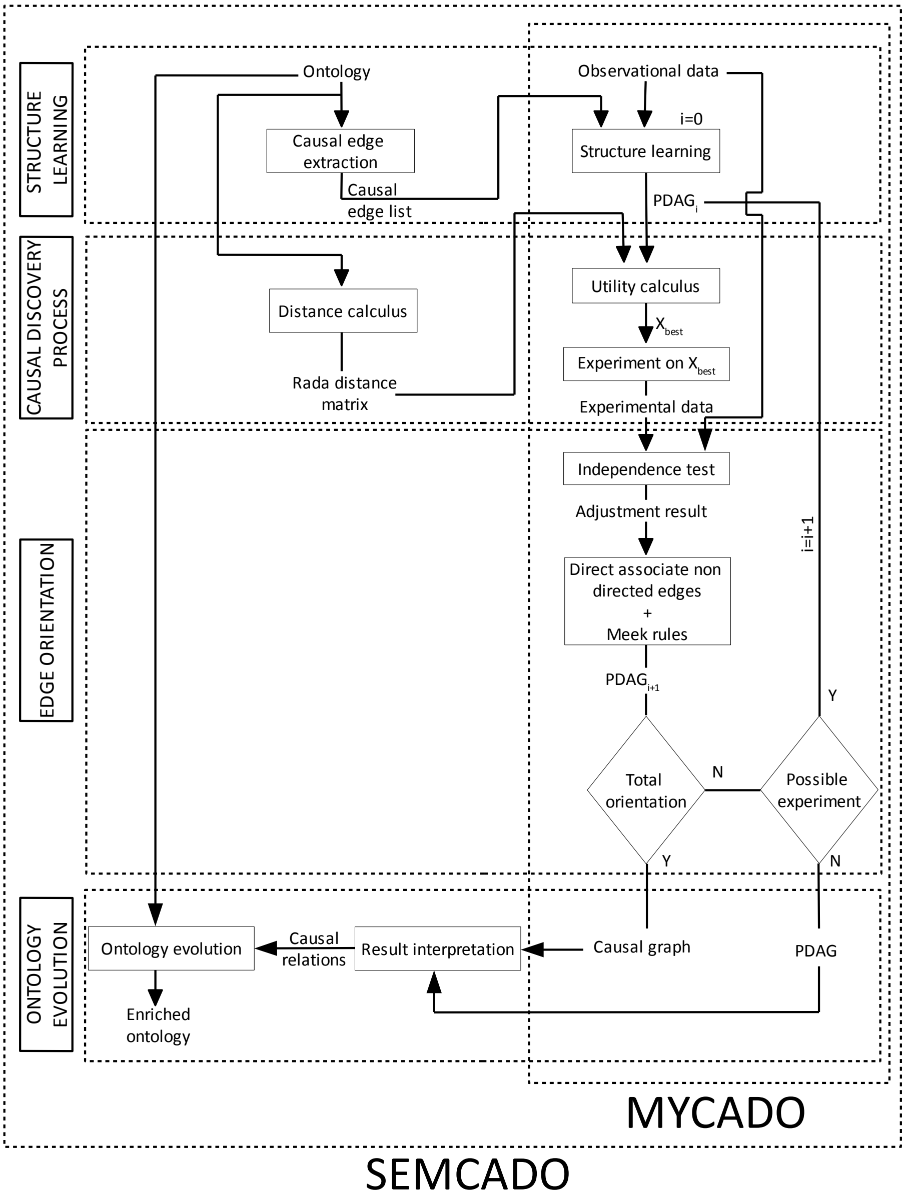


Figure 4.2: SemCaDo: Extending MyCaDo to allow CBN-Ontology interactions

Definition 4.1. *Given a domain ontology $\mathcal{O}$, let G=(C, R_c) be the DAG where R_c: $C_i \times C_j$ represents the subset of semantic causal relations extracted from $\mathcal{O}$. This subset included both direct and logically derivable semantic causal relations. Let H=(X, E_h) be a PDAG, where X is the set of the corresponding random variables and E_h corresponds to the causal dependencies between them. H is consistent with the existence restrictions in G if and only if:*

$\forall\ C_i,\ C_j \in C$, *if* $C_i \rightarrow C_j \in R_c$ *then* $X_i \rightarrow X_j \in E_h$.

When we are specifying the set of existence restrictions to be used, it is necessary to make sure that these restrictions can indeed be satisfied. In fact, such causal integration may lead to possible conflicts between the two models. When this occurs, we have to maintain the initial causal information in the PDAG since we are supposed to use perfect observational data. On the other hand, we should ensure the consistency of the existence restrictions in such a way that no directed cycles are created in G.

4.3.2 Causal discovery process

We start by deciding which experiment will be performed and hence also which variables will be altered and measured. For this purpose, a decision theoretic approach (i.e. Maximax, Maximin, Expected Utility) based on the ontological distance calculus [92] (See section 3.2.3) will be undertaken in order to guide the iterative causal discovery process and choose the more significant experimentations. By contrast, our strategy represents the exact opposite of much traditional experimental designs. Usually, the most studied concepts are the closest ones referring to the domain ontology. Here, on the contrary, we will advantage experimentations between the more distant concepts. By this way, we will accentuate the serendipitous aspect of the proposed strategy and investigate new and unexpected causal relations on the graph.

Moreover, distinctly to MYCADO algorithm and GES-EM adaptive approach (as described in sub-section 2.5.5) , the selection criterion used in the decision theoretic approach of SEMCADO is a semantical generalization of node connectivity.

Thus the utility function $U(.)$ will be an extension of Equation 2.13, by generalizing the first term $Ne_U(X_i)$ and replacing it by the semantical inertia, denoted by $SemIn(X_i)$ and expressed by:

$$SemIn(X_i) = \frac{\sum\limits_{X_j \in Nei(X_i) \cup X_i} dist_{Rada}(mscs(Nei^*(X_i) \cup X_i^*), X_j^*)}{Card(Nei(X_i) \cup X_i)} \tag{4.1}$$

where L^* represent the set of concepts relative to the set of nodes L, $mscs(L^*)$ is the most specific common subsumer of the set of concepts L^* and $dist_{Rada}(C_i, C_j)$ is the size of the shortest path between C_i and C_j.

The semantical inertia presents three major properties:

- When the experimented variable and all its neighbors lie at the same level in the concept hierarchy, the semantic inertia will be equal to the number of hierarchical levels needed to reach the mscs.
- If the corresponding concepts have the same parent in $\mathcal{H}$, then $SemIn$ will be proportional to $Card(Nei(.))$.
- It essentially depends on semantic distance between the studied concepts. This means that the more this distance is important, the more the $SemIn$ will be maximized.

Further to these, we also integrate a semantic cumulus relative to the inferred edges denoted by $Inferred\_Gain$ in our utility function. For this purpose, we use I$^*(X_i)$ to denote the set of concepts corresponding to nodes attached by inferred edges after performing an experimentation on X_i. So, the $Inferred\_Gain$ formula is expressed as follows:

$$Inferred\_Gain(X_i) = \frac{\sum\limits_{X_j \in I(X_i)} dist_{Rada}(mscs(I^*(X_i)), X_j^*)}{Card(I(X_i))} \tag{4.2}$$

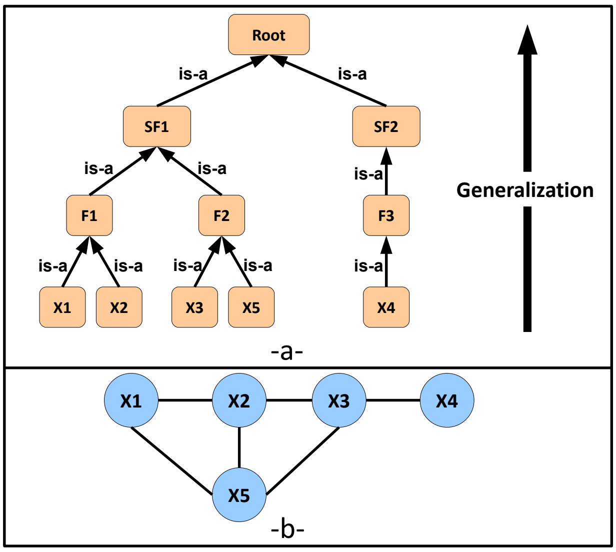


Figure 4.3: An illustration of is-a Tree (a) and the corresponding CPDAG (b)

$Inferred\_Gain$ also represents a generalization of $Card(inferred(inst(.))$ (refer to Equation 2.13) and depends on the semantic distance between the studied concepts.

When using the two proposed terms, our utility function will be as follows:

$$U(X_i) = \frac{SemIn(X_i) + Inferred\_Gain(X_i)}{\alpha.cost(A_{X_i}) + \beta.cost(M_{X_i})} \tag{4.3}$$

where the two measures of importance α, β are usually chosen proportional (α, $\beta \in$ [0,1] and max(α, β)$\notin$ 0).

Through this utility function, we provide a more explicit understanding that supports the desired effects of serendipitous revelation.

Example 4.1. *Given the domain ontology of figure 4.3.a, we analyze the semantical inertia of the two nodes X_2 and X_4 in the corresponding CPDAG presented in Figure 4.3.b.*

From the beginning, we show that the first node is more connected than the second so if we proceed with previous approaches, the selected node for experimentation is obviously X_2. But when proceeding with semantical iner-

tia, the choice of optimal variables to experiment on can change considerably since the connectivity is not all the time synonymous of higher Rada distance cumulus.

In what follows, the semantical inertia calculation details of the two nodes X_2 *and* X_4*:*
$Ne_U(X_2)$*=\{*X_1*,* X_3*,* X_5*\} ;*
$Ne_U(X_4)$*=\{*X_3*\};*
$mscs(X_2 \cup Ne_U(X_2))$*= \{*SF_1*\} ;*
$mscs(X_4 \cup Ne_U(X_4))$*= \{Root\};*
$SemIn(X_2)$*=(2+2+2+2)/4=2;*
$SemIn(X_4)$*=(3+3)/2=3;*

Hence, based on the SemIn criteria, experimenting on X_4 *is therefore more interesting than* X_2*.*

Now we will assume that we want to perform an action on the node X_1 *on the same CPDAG. An overview of all possible instantiations of the edge* $X_1 - Ne_U(X_1)$*, the possible structures compatible with each instantiation and edges susceptible to be inferred is given in Figure 4.4.*

According to those results, we will be able to find the utility of such action for each decision criteria. Here, we note that to simplify the calculations, we will consider equal costs for all decision criteria, namely, Cost(A_{X_1}*)+Cost(*M_{X_1}*)=1+2=3. In the third column of Figure 4.4, we represent associated Rada distance cumulus according to domain ontology as shown in Figure 4.3.a.*

Hence the three decision criteria will give us the following results:

- ***Maximax:*** *the maximum inferred cumulus is equal to 12, such that the utility for Maximax will be:*

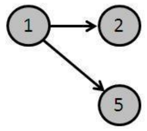
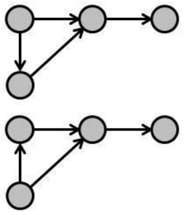
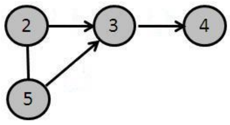
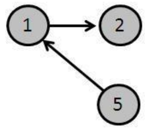
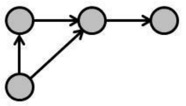
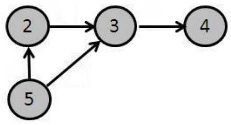
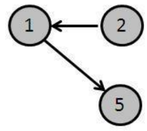
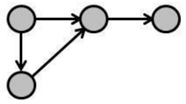
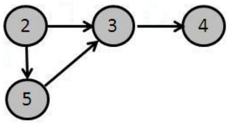
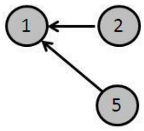
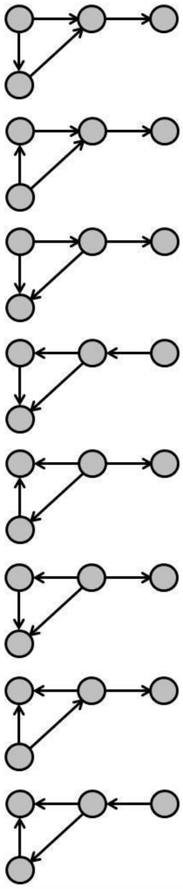
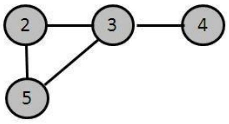
Instantiations	Possible structures	Result of edge inference	Inferred cumulus	P(inst)
			12	1/6
			12	1/12
			12	1/12
			0	2/3

Figure 4.4: All possible instantiations for X_i-$Ne_U(X_i)$, the possible structures compatible with each instantiation and the result of edge inference

$$\begin{aligned} U(X_1) &= \frac{SemIn(X_1) + \underset{inst(A_{X_1})}{max}(\#inferred\_cumulus(inst(A_{X_1})))}{cost(A_{X_1}) + cost(M_{X_1})} \\ &= \frac{2+12}{1+2} = \frac{14}{3} = 4.66 \end{aligned}$$

- ***Maximin:*** *the worst inferred cumulus is equal to 0 since we have an instantiation with no inferred edges. So the utility for Maximin is given by:*

$$\begin{aligned} U(X_1) &= \frac{SemIn(X_1) + \underset{inst(A_{X_1})}{min}(\#inferred\_cumulus(inst(A_{X_1})))}{cost(A_{X_1}) + cost(M_{X_1})} \\ &= \frac{2+0}{1+2} = \frac{2}{3} = 0.66 \end{aligned}$$

- ***Expected utility:***

  As shown in figure 4.4, there are 12 DAGs in the equivalence class of the example (all possible structures that can be inferred for all the instantiations). So the Expected utility for X_1 is given by:

$$\begin{aligned} U(X_1) &= \frac{SemIn(X_1) + \sum_{inst(A_{X_1})} \#inferred\_cumulus(inst(A_{X_1}))P_{eq}(inst(A_{X_1}))}{cost(A_{X_1}) + cost(M_{X_1})} \\ &= \frac{2 + (12 \times \frac{1}{6} + 12 \times \frac{1}{12} + 12 \times \frac{1}{12})}{1+2} = 2.33 \end{aligned}$$

For each decision criteria strategy, we have to calculate all node utilities and choose the best one in order to improve the causal discovery process.

In table 4.1, we compare the results of U(X_1) when applying SemCaDo (resp. MyCaDo) with the three decision criteria.

4.3.3 Edge orientation

Once the specified intervention performed, we follow the same edge orientation strategy as in Mycado [77]. Roughly speaking, the intervention takes place by means of mutilating all incoming arcs on the specified variable in

Strategy Decision criteria	MYCADO	SEMCADO
$MaxiMax$	2	4.66
$MaxiMin$	0.66	0.66
$ExpectedUtility$	0.78	2.33

Table 4.2: Comparing decision criteria in MYCADO and SEMCADO.

order to generate the experimental data. The obtained dataset as well as the initially supplied observations will be transferred to the chi2 adjustment test in order to determine if the variable experimented on is the cause or the effect of its neighboring variables.

Effectively, when experimenting on a variable X and measuring the effect on a neighboring variable Y, we have to determine if there is a significant association between the two rows data produced before and after the intervention on X.

Let $Nexp$ and $Nexp_{Y=y_i}$ be the total number of experimental data and the number of experiments where we obtain Y=y_i ($y_i \in D_Y$). The corresponding chi square statistic will be equal to:

$$\chi^2 = \frac{\sum_{i=1}^{|D_Y|} (Nexp_{(Y=y_i)} - (Nexp \times P(Y = y_i)))^2}{Nexp \times P(Y = y_i)} \tag{4.4}$$

Finally, by applying PC rules (see section 2.3.3), we can infer new undirected edges based on the experimentation edge orientation. If there are still some non-directed edges, we re-iterate over the second phase and so on, until no more causal discoveries can be made.

4.3.4 Ontology evolution

The causal knowledge will be then extracted from the CBN and interpreted for an eventual ontology evolution. More precisely, the causal relations will be traduced as semantic causal relations between the corresponding ontology concepts. We note that, because of the priority given to the ontology axioms (See subsection 4.2.1), only causal relations ensuring semantic consistency will be retained for the ontology evolution process. For this purpose, SemCaDo algorithm uses the six-phases evolution process (previously shown in Figure 3.4):

- *Change capturing:* the aim of this initial step in the ontology evolution process is to capture the new discovered causal relations on the current causal graph which are not actually modeled. It starts after obtaining a final causal structure in order to treat all changes in a consistent and unified manner.

- *Change representation:* in order to be correctly implemented, we have to represent these causal changes formally, explicitly and in a suitable format. In the context of SemCaDo algorithm, we only handle elementary changes [100] (i.e. restricted to adding semantic causal relations) that cannot be decomposed into simpler ones.

- *Semantics of change:* the semantics of change is the phase that enables the resolution of ontology changes in a systematic manner by ensuring the consistency of the ontology. In our case, conflicting knowledge is highly possible to occur when deducing causal conclusions from the ontology axioms. Such inconsistencies should be handled by automated reasoning. This step also prevents the creation of new cycles in the ontology when integrating the causal discoveries. This consistency rule is maintained since the causal discovery step in SemCaDo avoids the creation of cycles during the structure learning.

- *Implementation:* in order to avoid performing unwanted changes, a list of all consequences in the ontology and dependent artifacts should be generated and presented to the ontology engineer, who should then be able to accept the change or reject it. If the implementer agrees to add the new causal relationships, all actions to apply the change have to be performed.

- *Propagation:* pursuing and adopting the new causal discoveries can generate additional changes in the other parts of the ontology. These changes are called derived changes. That is why, during this step, it is necessary to determine the direct and indirect types of changes to be applied. In case of ambiguity, the ontology expert decides on the action to occur. A human intervention at this level is essential to remove the ambiguity and to make the final decision.

- *Validation:* change validation enables justification of performed changes and undoing them at user's request. If the output of SemCaDo causal discovery step is a partially directed graph, it is possible to restart the cycle when there is sufficient budget to make further discoveries.

During the causal discovery process, all experimentation results should be analyzed and interpreted in order to enrich domain ontology with new causal discoveries as detailed in section 3.3.

4.4 Conclusion

In this chapter, we give the the main correspondences that we made between the BNs and the ontologies in order to propose a decisional method for causal discovery and ontology evolution. We have also introduced the notion of serendipity which will be very useful when setting the experimental design.

The whole of the next chapter will be devoted to the discussion of experimental results that we obtained with both simulations and application on real system. Appendix C provides some of the implementation tricks that we used when developing and testing the SemCaDo approach.

A major challenge in computational biology is to uncover gene interactions
and key biological features of cellular systems.
Nir Friedman (2001)

Chapter 5

Experimental Study

5.1 Introduction

THe experimental study presented within this chapter covers two main levels, which are separate but related. First, we proceed through simulated causal networks and ontologies to demonstrate the efficiency of SemCaDo. Then we investigated the application of our approach to the problem of identifying the best experimental design when learning the gene regulatory circuitry from Saccharomyces cerevisiae cell cycle microarray data and Gene Ontology. In the remainder, we mainly focus on the application of SemCaDo to one biological task but this does not exclude, where appropriate, its application to other challenging modeling problems. All the implementations have been written in C++ using the API ProbT [1].

5.2 Validation through preliminary simulations

While it is important to study our algorithm's effectiveness in achieving its goal, it is also important to compare its performance with other algorithms designed for the same purpose. In our context, the MyCaDo algorithm is well suited for an experimental performance comparison with SemCaDo since it proposes a controlled experimentation's design and shares the same assumptions about the causal discovery process. A standard methodology

[1] http://www.probayes.com/

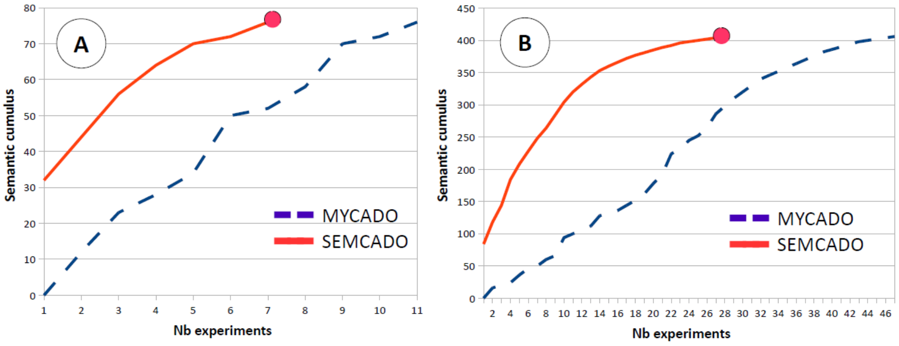


Figure 5.1: The semantic gain given the number of experiments using MyCaDo and SemCaDo on relatively small graphs (a) and bigger ones (b)

for evaluating the SemCaDo (resp. MyCaDo) performance is to proceed through simulations and evaluate the two algorithms in different test conditions.

5.2.1 Structure learning

First, a set of syntectic 50 and 200 node graphs are generated randomly from a uniform distribution. A DAG-to-CPDAG algorithm [19] is then applied on those CBNs in order to simulate the result of a structure learning algorithm working with a perfect infinite dataset. Then, for each simulated graph, we automatically generate a corresponding concept hierarchy after traversing the entire graph using the Depth-First Search algorithm. This method allows the obtention of a concept hierarchy with representative distances between the leaves according to the topological order. Finally, we use the initial causal graph to integrate a varying percentage (10% to 40%) of the causal relations in the ontology.

5.2.2 Causal discovery process

As we do not dispose of a real system to intervene upon, we decide to simulate the experimentations directly in the previously generated CBNs as in [77] and choose equal measures of importance when calculating the expected utilities (i.e. α=β=1). We also assume equal costs of intervention ($cost(A_{X_i})$=1) and measurement ($cost(M_{X_i})$= $\#Nei(X_i)$) and proceed using the MaxiMin decision criterion.

To perform the experimentation on the best node, we have to mutilate (i.e. disconnect) the node X_{best} from $Pa(X_{best})$ in the DAG such that the manipulated variable become totally independent of its parents in the post-intervention distribution [87]. We force X_{best} to take on random values and then sample the post-intervention distribution to get our experimental data. When determining if the experimented variable is the cause or the effect of its neighboring variables, we fix the significance level to 5%.

Another point to consider in our experimental study concerns the calculation of the semantic gain. In fact, after each SemCaDo (resp. MyCaDo) iteration, we measure the sum of Rada's distances [92] relative to the new directed edges in the graph and update a semantic gain:

$$Sem\_gain(X_i) = \sum_{X_j \in IN(Dir\_inf(X_i)),\ X_k \in OUT(Dir\_inf(X_i))} dist_{Rada}(X_j, X_k) \tag{5.1}$$

where: $Dir\_inf(X_i)$ represents the set of edges directly oriented or inferred after performing an experiment on X_i.
IN represents the set of $Dir\_inf()$ edge sources.
OUT represents the set of $Dir\_inf()$ edge destinations.

In both strategies, the two corresponding curves are increasing in, meaning that the higher is the number of experimented variables, the higher is the value of the semantic gain. Nevertheless the more the curve is increasing faster, the more the approach is converging to the best and most impressive experiments.

Figure 5.1 shows that, during the experimentation process, our approach comfortably outperforms the MyCaDo algorithm in term of semantic gain. This is essentially due to the initial causal knowledge integration and the causal discovery strategy when performing the experimentations. However the two curves reach a common semantic maxima when obtaining a fully directed graph. This is always the case since without using the same experimentations, the two strategies orient the same number of edges when finishing the experimental process. In this regard we should remember that we are approaching a decision problem which is subject to the experimentation cost and the budget allocation. Taking into account this constraint, the domination of SemCaDo will be extremely beneficial when the number of experiments is limited.

5.2.3 Ontology evolution

Finally, we have to reuse all these discovered causal edges to make the evolution of the joint ontology. However, this latter step is not so significant since we are generating the ontology from the simulated graphs.

5.3 Validation on S. cerevisiae cell cycle microarray data

Discovering and modeling gene regulatory circuitry from both observational and experimental data is one of the most challenging problems facing biologists today. This is essentially due to the non-negligible number, duration and cost of experiments and the lack of facilities for conducting genetic [2] (resp. environmental [3]) perturbations. In such circumstances, it would be far better to propose an experimental design to cope with the lack of data and provide maximal expected information. In this context, we propose to validate our approach using Saccharomyces cerevisiae cell cycle microarray data [97] and the corresponding Gene Ontology annotations.

[2] Gene knockout (deletion of the gene), or overexpression (setting the expression level higher than its usual level).

[3] change in one or more non-genetic factors, such as a change in environment, nutrition, pressure or temperature.

5.3.1 Molecular biology basics

The basic unit of structure and function in all self-replicating organisms is the cell. All biological information required for the functioning and development of a cell is encoded in the Deoxyribonucleic acid (DNA) sequence that is passed on from one cell to another in inheritance. The DNA sequence involves millions or even billions of nucleotide bases. These bases: Adenine (A), Guanine (G), Cytosine (C) and Thymine (T) are arranged in a specific order according to our genetic ancestry. Small fragments of DNA sequence encode the genes of an organism. Expression of the genes leads to formation of proteins. The synthesis procedure of most cellular organisms follows the central dogma: DNA → RNA → Protein product. The nucleotide sequence in a DNA (A, T, C, G) is first transcribed into another type of polymer called messenger Ribonucleic Acid (mRNA). mRNA is much smaller yet less stable than DNA. The nucleobase thymine (T) in a DNA is substituted by uracil (U) in an mRNA. After the transcription phase, mRNAs are spliced by removing the introns (i.e. sequences which do not encode proteins) and ligating together the separated exons (i.e. sequences encoding the same protein). A spliced mRNA is then translated into a protein by ribosome and transfer RNAs (tRNAs). The process of gene expression is used by all known life-eukaryotes (multicellular organisms) and prokaryote (bacteria) to generate the macromolecular machinery for life. The Human Genome project [64], one of the primary goals of which was to identify all protein coding genes, has estimated and identified approximately 20,000-25,000 genes in human DNA. Since the completion of the project, we have witnessed the emergence of various high-throughput technologies (such as DNA microarrays [14], protein arrays [73]). These technologies produce measurement data concerning the expression (or activity) of all genes in a genome simultaneously. Analysis of such measurement data requires the use of efficient and robust computational tools. The expected output from microarray analysis is a set of genes which are co or differentially expressed. Biological interpretation of such outcome is then necessary to investigate the mechanisms that cause such expression and improve our understanding of gene regulation.

5.3.2 Data description

The experimentation using real biological systems requires the use of gene-expression microarray data, the Gene Ontology and causal pathway repositories.

- ◇ *Gene expression dataset:* We consider the Yeast Saccharomyces cerevisiae cell cycle microarray data [97] since the Yeast genome is relatively small compared to more complex eukaryote organisms and highly annotated with Gene Ontology functions. In this dataset, the mRNA concentrations of nearly 6178 genes were measured with three independent fluorescence measurement methods. Overall, the data set contains 73 sampling points for all genes. Each of them is measured in different phases of the yeast cell cycle. According to [97], about 800 of these genes have been reported with varying transcripts over the cell cycle stages.

- ◇ *Gene ontology:* Most of the Saccharomyces cerevisiae genes are annotated with specific biological functions from the Gene Ontology [4] (GO), which remains the most popular initiative aiming at providing a structured, precisely defined, and dynamic controlled vocabulary to facilitate the description of gene roles and gene product attributes in the eukaryotic genome. The GO structure is in the form of a rooted DAG where nearly 30000 concepts are formalized into three related (sub-)ontologies, referred to as molecular function, cellular component and biological process (See Figure 5.2). According to the GO consortium, these GO domains represent three separate ontologies which are unrelated by a common parent node.

  The GO concepts are given a unique ID number in the form of GO:N (where N is a natural number) to identify and characterize some biological properties. Generally, the directed edges between concept nodes represent either subsumption links ("is-a") or composition relationships ("part of"). Nevertheless, another relationship can be found in the GO where one process (resp. function) directly affects another pro-

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

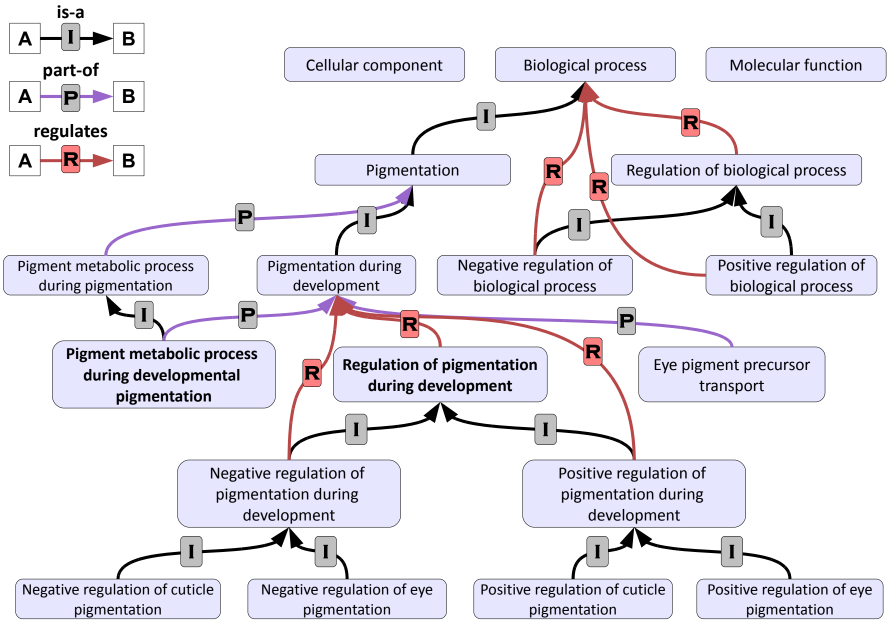
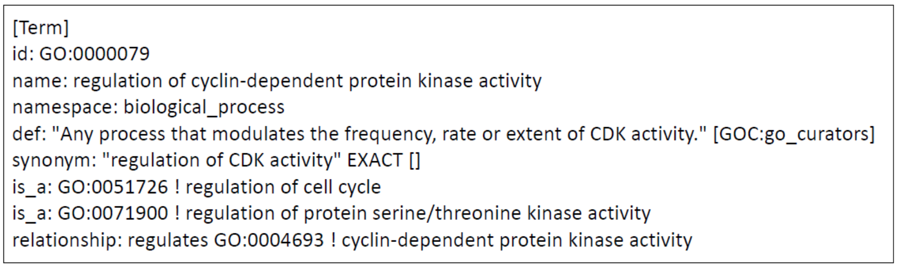

is-a
A
I
B
part-of
A
P
B
regulates
A
R
B
Cellular component
Biological process
Molecular function
Pigmentation
Regulation of biological process
Pigment metabolic process during pigmentation
Pigmentation during development
Negative regulation of biological process
Positive regulation of biological process
Pigment metabolic process during developmental pigmentation
Regulation of pigmentation during development
Eye pigment precursor transport
Negative regulation of pigmentation during development
Positive regulation of pigmentation during development
Negative regulation of cuticle pigmentation
Negative regulation of eye pigmentation
Positive regulation of cuticle pigmentation
Positive regulation of eye pigmentation

Figure 5.2: Screen capture from the GO

cess (resp. function), i.e. the former regulates the latter. For example, as it is depicted in Figure 5.2, "regulation of pigmentation during development" (GO:0048070) regulates "pigmentation during development" (GO:0048066). This "regulates" relationship especially covers processes, enzyme activities and molecular functions. So its semantics is quite different from the gene regulatory relations (i.e. fine-grained relations) that we try to discover using our approach. This GO structure (concepts + relationships) reflects the current representation of biological knowledge as well as serving as a guide for classifying new data. In Figure 5.3, we show an example of XML schema that identifies the GO:0000079 term. The set of tags indicate basic information such as: common name, corresponding sub-ontology, definition, synonym, derived is-a relations (resp. part-of relations) and derived regulatory relations.

According to the existing biomedical literature's assertions, the gene products may be annotated to as many GO concepts as needed, at the most specific levels possible. For instance, as shown in Figure 5.4, the



```
[Term]
id: GO:0000079
name: regulation of cyclin-dependent protein kinase activity
namespace: biological_process
def: "Any process that modulates the frequency, rate or extent of CDK activity." [GOC:go_curators]
synonym: "regulation of CDK activity" EXACT []
is_a: GO:0051726 ! regulation of cell cycle
is_a: GO:0071900 ! regulation of protein serine/threonine kinase activity
relationship: regulates GO:0004693 ! cyclin-dependent protein kinase activity
```

Figure 5.3: An example of GO term identification in XML format.

gene CLB6 is involved in:

1. the regulation of cyclin-dependent protein kinase activity (GO:0000079),
2. the regulation of S phase of mitotic cell cycle (GO:0007090),
3. the G1/S transition of mitotic cell cycle (GO:0000082).

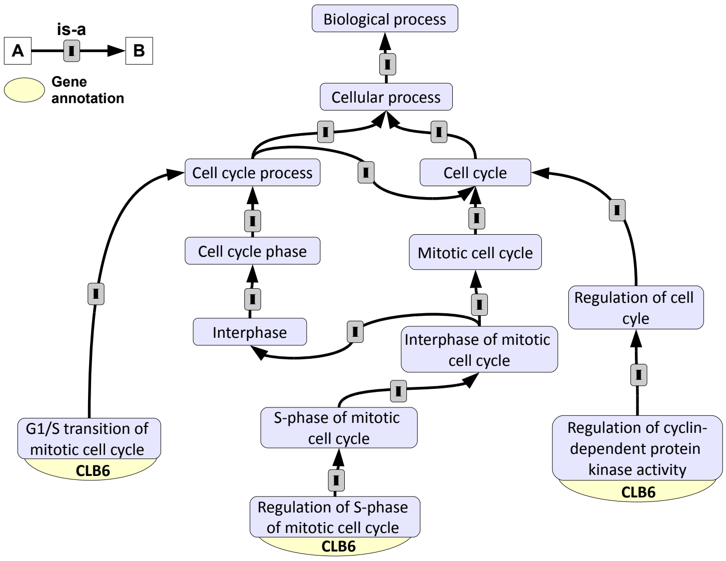

is-a
A
B
Gene annotation
Biological process
Cellular process
Cell cycle process
Cell cycle
Cell cycle phase
Mitotic cell cycle
Regulation of cell cyle
Interphase
Interphase of mitotic cell cycle
G1/S transition of mitotic cell cycle
CLB6
S-phase of mitotic cell cycle
Regulation of cyclin-dependent protein kinase activity
CLB6
Regulation of S-phase of mitotic cell cycle
CLB6

Figure 5.4: CLB6 multiple localizations in GO

However, many other genes are not annotated at a particular level of the GO due to the lack of available biological information or GO incompleteness. Such a classification will provide a higher-level un-

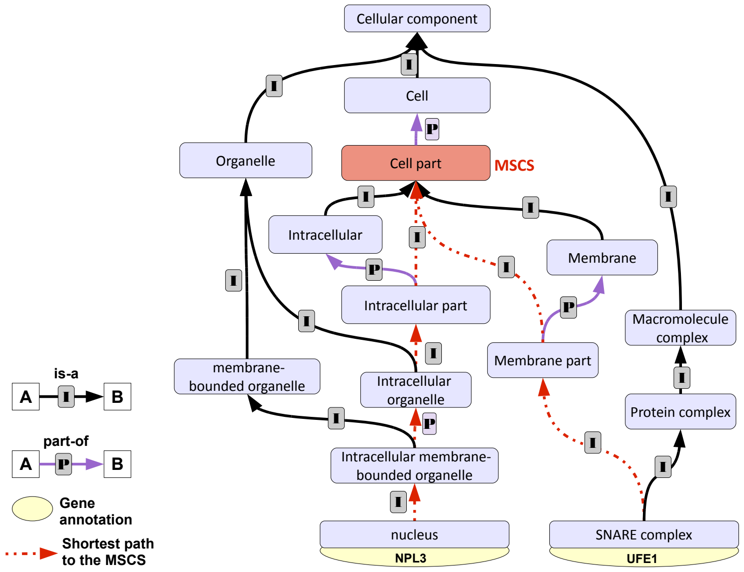


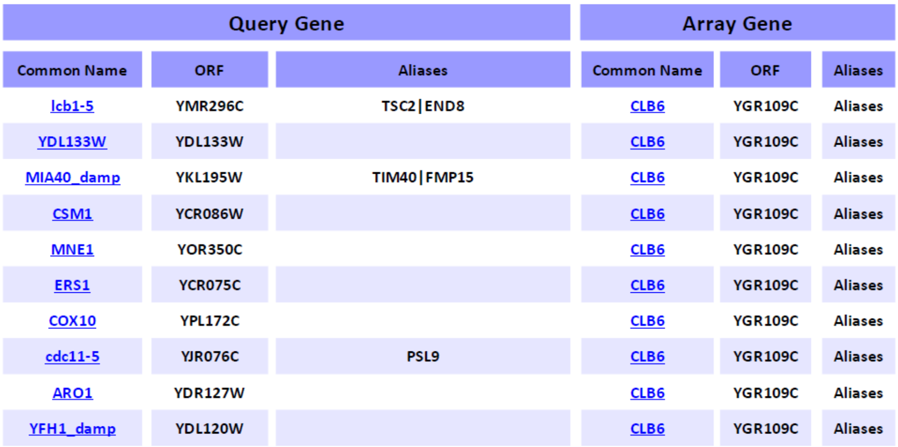
Figure 5.5: Semantic distance between two annotated genes in GO.

derstanding of how tissue-specific genes are regulated and biologically expressed.

Given two other genes NPL3 and UFE1 which are respectively annotated with the cell nucleus (GO:0005634) and the SNARE complex (GO:0031201), we show in Figure 5.5 the multiple paths that can be found between them. Using our simple path based method, we set the cell part term (colored in red) as the mscs of the two studied concepts. If there are multiple paths between any two concepts and their mscs, only the shortest one is considered. The red dashed lines indicate in our case the optimal path according to the GO structure. We note that the best GO-distance between two genes can be equal to 0 when both of them are annotated to the same GO concept.

◇ *Causal pathway repositories:*

However, since the GO structure consists essentially of hierarchical classification, we will be unable to extract or enrich the GO with regulatory pathways. An alternative way to identify causal relations is to use the so-called Biochemical Pathway Repositories where regulatory



Query Gene			Array Gene		
Common Name	ORF	Aliases	Common Name	ORF	Aliases
lcb1-5	YMR296C	TSC2|END8	CLB6	YGR109C	Aliases
YDL133W	YDL133W		CLB6	YGR109C	Aliases
MIA40_damp	YKL195W	TIM40|FMP15	CLB6	YGR109C	Aliases
CSM1	YCR086W		CLB6	YGR109C	Aliases
MNE1	YOR350C		CLB6	YGR109C	Aliases
ERS1	YCR075C		CLB6	YGR109C	Aliases
COX10	YPL172C		CLB6	YGR109C	Aliases
cdc11-5	YJR076C	PSL9	CLB6	YGR109C	Aliases
ARO1	YDR127W		CLB6	YGR109C	Aliases
YFH1_damp	YDL120W		CLB6	YGR109C	Aliases

Figure 5.6: Screen capture of the top DRYGIN regulatory pathways involving the gene CLB6.

information could be available. Fueled by the availability of experimentally determined pairwise gene interactions, different datasets for delineating the biochemical pathways and reactions have been merged. Most of these scientific databases such as, Data Repository of Yeast Genetic Interactions (DRYGIN) [5] [60], enable a convenient access to genes in terms of the biological pathways in which they intervene [4] (See the DRYGIN screen capture for the top regulatory pathways involving the gene CLB6 in Figure 5.6).

5.3.3 Experimental design

Table 5.1: The set of all possible correspondences between the GRN and the Gene Ontology.

Gene Regulatory Network	Gene Ontology
Nodes	Concept instances (i.e. GO annotations)
Causal dependencies	Semantic causal relations
Causal inference	Logic rule reasoning

[5] http://drygin.ccbr.utoronto.ca/

When applying our approach in the context of biological field, we were forced to change some of the initial CBN-ontology correspondences that we provide in subsection 4.2.2. According to Table 5.1, the GRN nodes which correspond to genes will be assigned to the most specific level of the Gene Ontology using term annotations (i.e. instances). Then there would no longer be any need to use the observational and experimental data since we dispose of an appropriate causal model based on we simulate experimental treatments. Finally, the causal inference in the GRN will be assigned to the GO logic rule reasoning[6].

To make a meaningful performance comparison between MyCaDo and SemCaDo algorithms, we will detail the three main blocs of our experimental strategy (refer to figure 4.2 in sub-section 4.3 to follow the cycle in more details):

- ◇ *Structure learning:* Our alternative way for implementing the MyCaDo (resp. SemCaDo) approaches is to use the Gene Regulatory Network (GRN) of [40] as a starting causal model and the GO structure as a source for calculating semantic distances between genes. From a modeling standpoint, a GRN can be thought as a DAG $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ where $\mathcal{V}$ is the set of n gene nodes (resp. protein concentrations and other experimental conditions) and $\mathcal{E}$ is the set of directed edges among the nodes belonging to $\mathcal{V}$. Such models are well suited for representing cellular processes (i.e. metabolism, signal transduction and transport).

  Using the Yeast Saccharomyces cerevisiae cell cycle microarray data [97], [40] proved that they were able to extract a finer structure of regulatory interactions between genes. Their heuristic approach was aimed at focusing on a pair of features that are common to high-scoring networks. The first type of features they identified is the high confidence Markov relations which assumes that a gene interaction exists between two genes if no variable in the model mediates the dependence between them. The second feature is synonymous of causality in the model since, out of all 800 genes they treat, only a few seem to dominate the order (i.e., appear before many other genes) in the

[6] refer to Appendix B for additional details.

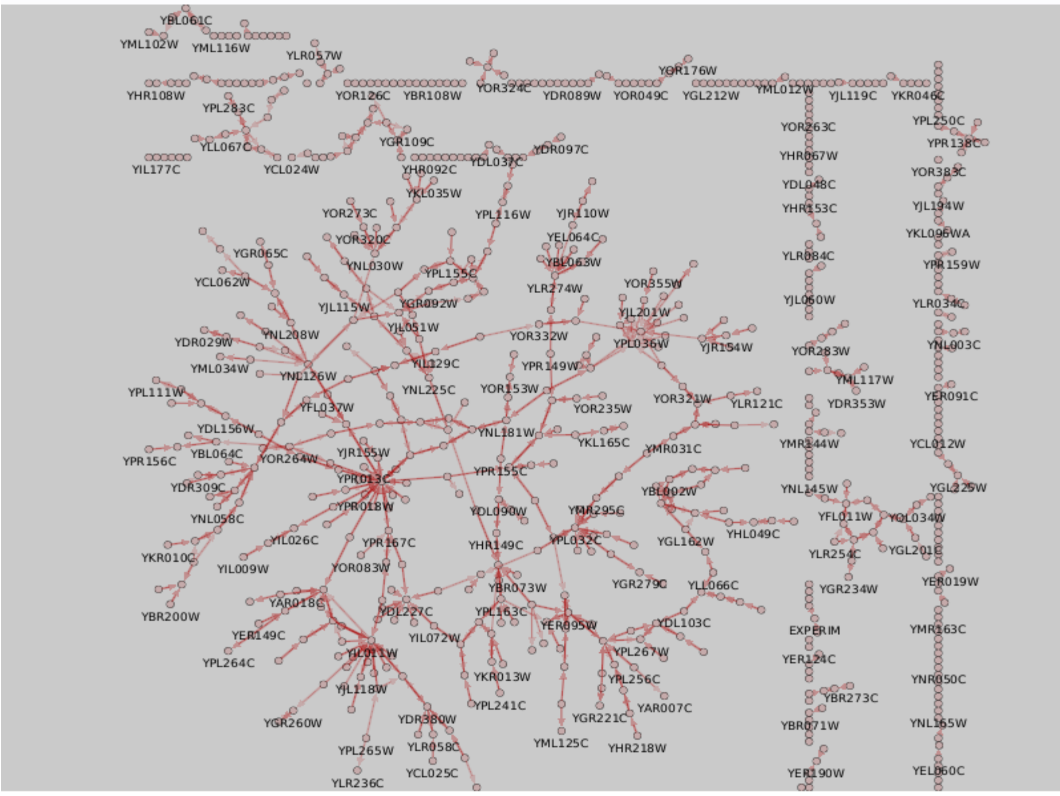


Figure 5.7: Graphical representation of the entire GRN [40] employed for the experimentations

overall networks of a given equivalence class . The intuition is that precedence over the ordering is indicative of potential cause-to-effect relationships on the cell-cycle process. Using the Tulip Software [3], a screen capture of the considered causal graph is shown in Figure 5.7.

The main reason for choosing the GRN of [40] is that it is free from assumptions and don't reuse any prior knowledge. We also note that interactions between genes other than causal relationships (i.e. directed edges with sufficiently high confidence in the order between genes) are not considered. Using this causal model, we apply the DAG-to-CPDAG algorithm [19] to extract the essential graph where from a total of 650 arcs we found 250 which become undirected. As a check of the effect of prior knowledge, we also consider three other cases in which we orient 10 % (resp. 20 and 30 %) of these undirected edges before starting the SemCaDo causal discovery.

- *Causal discovery process:* When calculating the SemCaDo utilities, we were also forced to add a "fake" term (GO root) as a parent of the three existing root nodes in the GO (i.e. molecular function, cellular component and biological process) to perform semantic distance calculations on one unique ontology. This GO root will be then associated with a dozen of S. cerevisiae gene products which are not yet annotated with any GO term. The rest of the experimental process remains unchanged from that used in subsection 5.2.

- *Pathway repository evolution:* Although, to make the experimental design more realistic in the context of biological resource management, we need to modify the third phase of our algorithm by updating the biological pathway datasets (e.g. DRYGIN repository) instead of making the GO enrichment. Metabolic pathways in such databases are computationally predicted using automated literature mining and then manually reviewed to ensure higher accuracy. This new dimension ensures optimal reuse of causal discoveries obtained from experimentations by submitting missing gene pairwise interactions. Unfortunately, since we are not intervening on a real system, we are unable to provide the dataset curators with any suggestions or corrections. We therefore content ourselves with a brief outline of the principle.

5.3.4 Results & interpretation

The corresponding results are reported in Figure 5.8 under four different test conditions. Each graphic displayed the evolution of the semantic cumulus along the experimental process for both MyCaDo and SemCaDo. This way of measuring the performance of the two methods is quite original since professionals from the biotechnology field often assume that functionally proximal genes or proteins are likely to interact with each other [82]. Here we would like to propose a different approach whose aim is to promote the experimentation on the more distant genes according to the GO. Table 5.2 can be used

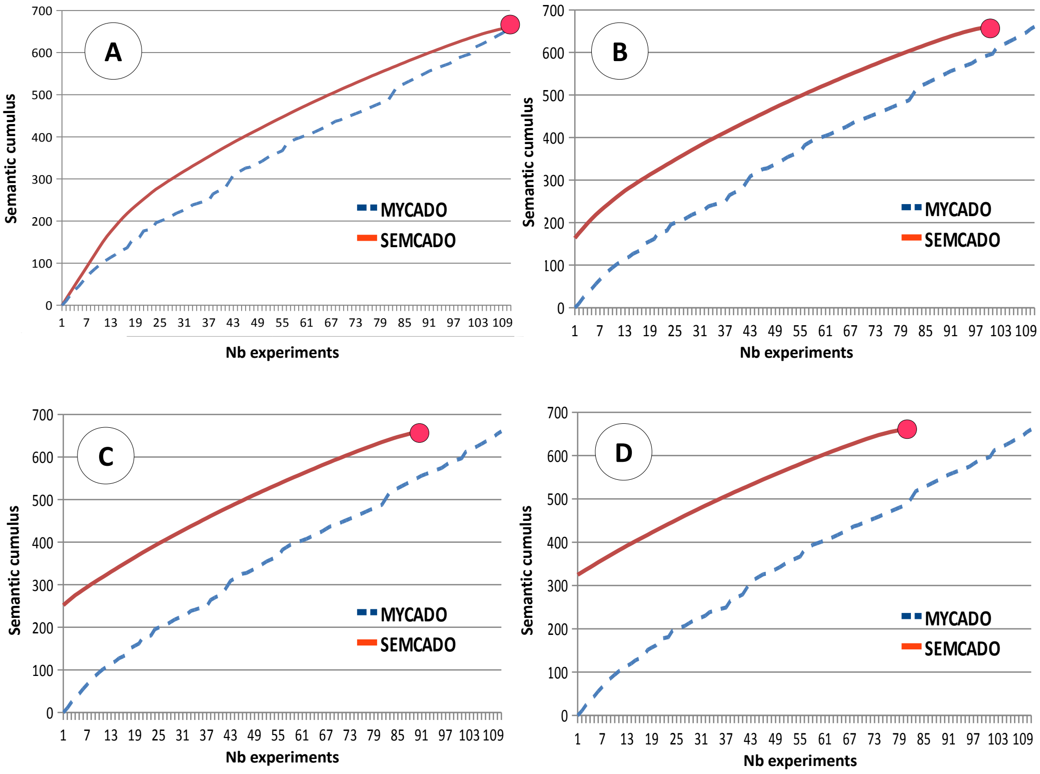


Figure 5.8: Comparison between MyCaDo and SemCaDo without any prior knowledge (a) and after integrating 10 %, resp. 20 and 30% (b, c, d).

in conjunction with Figure 5.8 to obtain additional statistical information relative to the gain in cumulus margin, the difference between the two curves areas and the number experiments that we saved when applying SemCaDo.

First of all, we apply both MyCaDo and SemCaDo without any prior knowledge (See Figure 5.8.a). The difference in areas between the two curves was about 13% and around one hundred experiments have been realized with the two algorithms. When we integrate 10% of the initial causal relations before starting the learning process (Figure 5.8.b) , we earned a cumulus margin of about 24% from the beginning. The difference in areas pass to 38% and we save nearly one dozen of experiments. This increasing trend continues when incorporating 20% of the initial causal relations (Figure 5.8.c) to obtain 38% as a cumulus margin, 44% as total difference in areas between the two curves and 24 less experiments. We finish with the integration of 30% of the initial causal relations (Figure 5.8.d) to reach a cumulus margin of about 45%, a total difference in areas between the two curves exceeding the 48% and save more than 30 unnecessary experiments.

Table 5.2: Statistical analysis of Figure 5.8.

Causal integration	Cumulus gain	Diff. curves areas	Nb. of saved experiments
0%	0%	13%	0
10%	24%	38%	12
20%	38%	44%	24
30%	45%	48%	30

From all those graphics, it is obvious that the integration of causal prior knowledge in the pathway modeling have greatly increased the reliability of SemCaDo in the GRN construction. A lot of experiments and efforts have been saved compared to MyCaDo and the most informative interventions have been reported earlier in the experimental process. This allows a significant gain in term of relevant experimentations especially when there is not enough budgets to cover all the required interventions. Our belief is that SemCaDo top-ranked genes can be targets for medical treatment of genetic diseases and opportunities to obtain further knowledge about the biological mechanisms that underlie their gene expression. Potentially, this gives us scope to explore virgin areas when developing our knowledge-base on pathway modeling.

5.4 Conclusion

The experimental results, provided in this chapter, show that the proposed algorithm achieves better performance than MyCaDo, its competing algorithm. The proposed approach was tested through simulations and then validated on a real system (S. Cerevisiae cell cycle microarray data) using the Gene Ontology to make gene pathway discoveries. Nevertheless, the main problem is that there is no commonly accepted benchmark to help us to go further towards developing experimental tests that can lead to more rigorous results.

Chapter 6

Conclusion

6.1 Summary

With the rising need to reuse the existing knowledge when learning CBNs, the ontologies can supply valuable semantic information to make further interesting discoveries with the minimum expected cost and effort.

In this thesis, we propose a cyclic approach in which we make use of the ontology in an interchangeable way. The first direction involves the integration of semantic knowledge to anticipate the optimal choice of experimentations via a serendipitous causal discovery strategy. The second complementary direction concerns an enrichment process by which it will be possible to reuse these causal discoveries, support the evolving character of the semantic background and make an ontology evolution.

To our knowledge, ours is the first attempt to design a two-way approach for coupling both probabilistic causal networks learning and ontological background.

Compared to MyCaDo, the experimental results obtained from different model simulations are very promising. The SemCaDo performance domination is reinforced through the validation on S. cerevisiae cell cycle microarray data to learn Gene Regulatory pathways using the Gene Ontology.

6.2 Advantages

Our new framework has several advantages over existing experimental design techniques. First, the idea of reusing ontological components can help to tackle real world learning problems.

So, instead of repeating the effort that have already been spent elsewhere to capture and create the same causal knowledge, one may reuse an existing domain ontology or some parts of it and make a considerable saving in term of time and cost. With such approach, we can also increase the reliability of the domain ontology by giving indication that it is continuously revised and evaluated through our ontology evolution process.

Moreover, the serendipitous aspect when choosing the experimentations to perform is another advantage of the proposed strategy. This allows us to discover virgin areas and move away from what it is usually proposed by the research community.

6.3 Applications

The results of this thesis are relevant to all communities dealing with Causal Bayesian Networks and disposing of a corresponding domain ontology. In chapter 5, we conducted an experimental study in the biological field and tried to learn causal regulatory pathways using the Gene Ontology. We can imagine innumerable uses for SemCaDo, some more obvious than others. We therefore outline potentially fruitful areas that can adopt a similar serendipitous experimental design.

- Chemistry: mineral processing, experimenting on acids and bases.
- Physics: potential application in the engineering sectors,
- Psychology: there is a real need to underly causes of behavior by studying humans and animals.
- Ecology: reveal the relationships between the organisms and the environmental factors.

- Marketing: can help marketing executives analyze how the various components of a marketing campaign influence consumer behavior.
- Health care: carefully design series of experiments to optimize medical devices and drug formulations.

6.4 Limitations

Despite our multiple attempts to take into account all interactions that can occur between the CBN and domain ontology, we are still making strong assumptions when designing our SemCaDo approach. For example, when adopting the causal sufficiency assumption, we eliminate a number of latent variables that can be part of the model to pick up. Those hidden variables can be of particular relevance to establish useful models for achieving a correct causal inference and predicting the effects of some external criteria.

The second limitation occurs in the ontology evolution process because of the priority given to the ontology axioms. So in each SemCaDo iteration, we are obligated to retain only causal relations which ensure the semantic consistency with the domain ontology and to throw away the potential opportunities to make the ontology revolution.

Finally, when dealing with domain ontology, a unique concept-attribute is considered when investigating cause-to-effect relationships. Unfortunately, with such strategy, we ignore many other concept-attributes that can be fruitfully exploited in our approach.

6.5 Issues for Future Research

Our framework offers several opportunities for future research, among them the expansion for better interactions with the ontology axioms during the causal discovery process, the use of other types of semantic relations and the integration of probabilities in OWL ontologies.

- Ontology revolution:

When making the ontology revolution, we accept that some of the previous ontology axioms become inconsistent with the new ontology version. Compared to ontology evolution, the scientific researches that treat the ontology revolution are quite rare and until now the technical specifications are not enough well defined. Therefore, such research issue can be of great interest to better investigate the possible ways to adapt ontology axioms according to the causal discoveries made during the learning process.

- Generalization to other types of semantic relations:

  Another important area of investigation concerns the generalization of the semantic causal relations. Let us remember that we adopt causal relations in the ontology when we detect some form of cause-to-effect relationships between shared concept-attributes. Topics for discussions will include how to generalize fine-grained causal relations to more generic forms of semantic relations between concepts.

  Moreover, the restriction to only taxonomic and semantic causal relations can be also relaxed to include other types of relationships. This requires more specificity in term of CBN-ontology correspondences and a detailed study to justify why and how they can contribute to the causal discovery process.

- Probabilistic ontologies:

  The next generation of knowledge-based systems needs to tap into large domain-specific knowledge and combine various modeling formalisms. Thus, the subsequent goal of coupling causal bayesian networks and ontologies is to propose a derived formalism combining the power of probabilistic (resp. causal) reasoning and ontology semantics. We can expand our interaction model to simulate more sophisticated coordination in order to obtain real probabilistic ontologies augmented with a powerful inference mechanism.

Bibliography

[1] Paul André, Jaime Teevan, and Susan T. Dumais. From x-rays to silly putty via uranus: serendipity and its role in web search. In Dan R. Olsen Jr., Richard B. Arthur, Ken Hinckley, Meredith Ringel Morris, Scott E. Hudson, and Saul Greenberg, editors, *CHI*, pages 2033–2036. ACM, 2009.

[2] Michael Ashburner, Catherine A. Ball, Judith A. Blake, David Botstein, Heather Butler, J. Michael Cherry, Allan P. Davis, Kara Dolinskiand Selina S. Dwight, Janan T. Eppig, Midori A. Harris, David P. Hill, Laurie Issel-Tarver, Andrew Kasarskis, Suzanna Lewis, John C. Matese, Joel E. Richardson, Martin Ringwald, Gerald M. Rubin, and Gavin Sherlock. Gene ontology: tool for the unification of biology. the gene ontology consortium. pages 25–29, 2000.

[3] David Auber. Tulip : A huge graph visualisation framework. In P. Mutzel and M. Jünger, editors, *Graph Drawing Softwares*, Mathematics and Visualization, pages 105–126. Springer-Verlag, 2003.

[4] Gary D. Bader, Michael P. Cary, and Chris Sander. Pathguide: a pathway resource list. *Nucleic Acids Research*, 34(Database-Issue):504–506, 2006.

[5] Montassar Ben Messaoud, Philippe Leray, and Nahla Ben Amor. Integrating ontological knowledge for iterative causal discovery and visualization. In *ECSQARU'09*, pages 168–179, 2009.

[6] Montassar Ben Messaoud, Philippe Leray, and Nahla Ben Amor. Semcado : a serendipitous strategy for learning causal bayesian networks using ontologies. In *ECSQARU'11*, pages 182–193, 2011.

[7] Montassar Ben Messaoud, Philippe Leray, and Nahla Ben Amor. Semcado: a serendipitous causal discovery algorithm for ontology evolution. In *The IJCAI-11 Workshop on Automated Reasoning about Context and Ontology Evolution (ARCOE-11), Barcelona, Spain*, pages 43–47, 2011.

[8] Emmanuel Blanchard, Mounira Harzallah, Henri Briand, and Pascale 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.

[9] Hanen Borchani, Nahla Ben Amor, and Khaled Mellouli. Learning bayesian network equivalence classes from incomplete data. In *Discovery Science*, pages 291–295, 2006.

[10] Hanen Borchani, Maher Chaouachi, and Nahla Ben Amor. Learning causal bayesian networks from incomplete observational data and interventions. *In proceedings of Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, 2007.

[11] Mark E. Borsuk, Craig A. Stow, and Kenneth H. Reckhow. Ecological prediction using causal bayesian networks: A case study of eutrophication management in the neuse river estuary. *Science And Technology*, 2002.

[12] Pierre Bourque, Robert Dupuis, Alain Abran, James W. Moore, and Leonard L. Tripp. The guide to the software engineering body of knowledge. *IEEE Software*, 16(6):35–44, 1999.

[13] Jacquelyn Burkell, Anabel Quan-Haase, and Victoria L. Rubin. Promoting serendipity online: recommendations for tool design. In Jens-Erik Mai, editor, *iConference*, pages 525–526. ACM, 2012.

[14] Alberto Calvi, Pietro Lovato, Simone Marchesini, Barbara Oliboni, Massimo Delledonne, and Alberto Ferrarini. Microarray system - a system for managing data produced by dna-microarray experiments. In Marco Pellegrini, Ana L. N. Fred, Joaquim Filipe, and Hugo Gamboa, editors, *BIOINFORMATICS*, pages 293–296. SciTePress, 2011.

[15] Rommel Novaes Carvalho. *Probabilistic ontology: Representation and modeling methodology.* PhD thesis, George Mason University, 2011.

[16] Silvana Castano, Alfio Ferrara, and Guillermo Nudelman Hess. Discovery-driven ontology evolution. In Giovanni Tummarello, Paolo Bouquet, and Oreste Signore, editors, *SWAP*, volume 201 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2006.

[17] Jie Cheng, David A. Bell, and Weiru 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.

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

[19] David M. Chickering. Learning equivalence classes of bayesian-network structures. *Journal of Machine Learning Research*, 2:445–498, 2002.

[20] David M. Chickering, Dan Geiger, and David Heckerman. Learning bayesian networks is np-hard. Technical report, Technical Report MSR-TR-94-17, Microsoft Research Technical Report, 1994.

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

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

[23] Gregory F. Cooper and Changwon Yoo. Causal discovery from a mixture of experimental and observational data. In Kathryn B. Laskey and Henri Prade, editors, *UAI*, pages 116–125. Morgan Kaufmann, 1999.

[24] Luis M. de Campos. A scoring function for learning bayesian networks based on mutual information and conditional independence tests. *J. Mach. Learn. Res.*, 7:2149–2187, 2006.

[25] Luis M. de Campos and Javier G. Castellano. Bayesian network learning algorithms using structural restrictions. *International Journal of Approximate Reasoning*, pages 233–254, 2007.

[26] Ann Devitt, Boris Danev, and Katarina Matusikova. Constructing bayesian networks automatically using ontologies. In *Second Workshop on Formal Ontologies Meet Industry. FOMI '06*, Trento, Italy, 2006.

[27] Zhongli Ding and Yun Peng. A probabilistic extension to ontology language owl. In *In Proceedings of the 37th Hawaii International Conference On System Sciences (HICSS-37), Big Island*, 2004.

[28] Zhongli Ding, Yun Peng, Rong Pan, and Yang Yu. A bayesian methodology towards automatic ontology mapping. In *Proceedings of the AAAI-05 C&O Workshop on Contexts and Ontologies: Theory, Practice and Applications*, 2005.

[29] Mareck J. Druzdel, Linda C. Van der Gaag, Max Henrion, and Finn Jensen. Building probabilistic networks: Where do the numbers come from? guest editors introduction. *IEEE Transactions on Knowledge and Data Engineering*, 12, 2000.

[30] Marek J. Druzdzel and Henri J. Suermondt. Relevance in probabilistic models: "backyards" in a "small world". In *In Working notes of the AAAI–1994 Fall Symposium Series: Relevance*, pages 60–63, 1994.

[31] Frederick Eberhardt. *Causation and Intervention.* PhD thesis, Department of Philosophy, Carnegie Mellon University, 2007.

[32] Frederick Eberhardt. Causal discovery as a game. *Journal of Machine Learning Research - Proceedings Track*, 6:87–96, 2010.

[33] Frederick Eberhardt, Clark Glymour, and Richard Scheines. N - 1 experiments suffice to determine the causal relations among N variables. *In Department of Philosophy, Carnegie Mellon University, Technical Report CMU-PHIL-161*, 2004.

[34] Frederick Eberhardt, Clark Glymour, and Richard Scheines. On the number of experiments sufficient and in the worst case necessary to identify all causal relations among n variables. In *UAI*, pages 178–184. AUAI Press, 2005.

[35] Xiangyu Fan, Javed Mostafa, Ketan Mane, and Cassidy R. Sugimoto. Personalization is not a panacea: balancing serendipity and personalization in medical news content delivery. In Gang Luo, Jiming Liu, and Christopher C. Yang, editors, *IHI*, pages 709–714. ACM, 2012.

[36] Stefan Fenz, A Min Tjoa, and Marcus Hudec. Ontology-based generation of Bayesian networks. In *International Conference on Complex, Intelligent and Software Intensive Systems, 2009. CISIS '09.*, pages 712–717, 2009.

[37] Ronald A. Fisher. *The design of experiments.* Hafner, 1935.

[38] Nir Friedman. Being bayesian about network structure. In *Machine Learning*, pages 201–210, 2000.

[39] Nir Friedman, Lise Getoor, Daphne Koller, and Avi Pfeffer. Learning probabilistic relational models. In Thomas Dean, editor, *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1300–1309, San Francisco, CA, 1999. Morgan Kaufmann.

[40] Nir Friedman, Michal Linial, and Iftach Nachman. Using bayesian networks to analyze expression data. *Journal of Computational Biology*, 7:601–620, 2000.

[41] Nir Friedman, Michal Linial, Iftach Nachman, and Dana Pe'er. Using bayesian networks to analyze expression data. *Journal of Computational Biology*, 7(3-4):601–620, 2000.

[42] Nir Friedman, Iftach Nachman, and Dana Pe'er. Learning bayesian network structure from massive datasets: The "sparse candidate" algorithm. In Kathryn B. Laskey and Henri Prade, editors, *UAI*, pages 206–215. Morgan Kaufmann, 1999.

[43] Barbara Furletti and Franco Turini. Knowledge discovery in ontologies. *Intelligent Data Analysis*, 16(3):513–534, 2012.

[44] Aldo Gangemi, Domenico Pisanelli, and Geri Steve. Ontology alignement: Experiences with medical terminologies. *In N. Guarino (ed.) Formal Ontology in Information Systems*, pages 163–178, 1998.

[45] Clark Glymour and G. Cooper. *Computation, Causation, and Discovery.* MIT Press, Cambridge, MA, 1999.

[46] Thomas L. Griffiths and Joshua B. Tenenbaum. Structure and strength in causal induction. *Cognitive Psychology*, 51:334–384, 2005.

[47] T. R. Gruber. Towards Principles for the Design of Ontologies Used for Knowledge Sharing. *International Journal Human-Computer Studies Vol. 43, Issues 5-6*, pages 907–928, November, 1995.

[48] Peter Haase and Ljiljana Stojanovic. Consistent evolution of owl ontologies. In *Proceedings of the Second European Semantic Web Conference, Heraklion, Greece*, 2005.

[49] Peter Haase and York Sure. State of the art on ontology evolution. SEKT Deliverable, 2004.

[50] Yang-Bo He and Zhi Geng. Active learning of causal networks with intervention experiments and optimal designs. *Journal of Machine Learning Research*, 9:2523–2547, 2008.

[51] David Heckerman. A tutorial on learning with bayesian networks. Technical report, Learning in Graphical Models, 1995.

[52] David Heckerman and David M. Chickering. Learning bayesian networks: The combination of knowledge and statistical data. In *Machine Learning*, pages 20–197, 1995.

[53] Jeff Heflin, James Hendler, and Sean Luke. Coping with changing ontologies in a distributed environment. In *In Proceedings of AAAI-99 Workshop on Ontology Management*, pages 74–79. Press, 1999.

[54] Markus Holi and Eero Hyvönen. A method for modeling uncertainty in semantic web taxonomies, 2004.

[55] Finn V. Jensen. *An Introduction to Bayesian Networks.* Springer Verlag, New York, 1996.

[56] Beom-Jun Jeon and In-Young Ko. Ontology-based semi-automatic construction of bayesian network models for diagnosing diseases in e-health applications. In *FBIT*, pages 595–602, 2007.

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

[58] Michael I. Jordan. *Learning in graphical models.* MIT Press, Cambridge, MA, USA, 1999.

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

[60] Judice L. Y. Koh, Huiming Ding, Michael Costanzo, Anastasia Baryshnikova, Kiana Toufighi, Gary D. Bader, Chad L. Myers, Brenda J. Andrews, and Charles Boone. Drygin: a database of quantitative genetic interaction networks in yeast. *Nucleic Acids Research*, 38(Database-Issue):502–507, 2010.

[61] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models - Principles and Techniques.* MIT Press, 2009.

[62] Kevin B. Korb, Lucas R. Hope, Ann E. Nicholson, and Karl Axnick. Varieties of causal intervention. In Chengqi Zhang, Hans W. Guesgen, and Wai-Kiang Yeap, editors, *PRICAI*, volume 3157 of *Lecture Notes in Computer Science*, pages 322–331. Springer, 2004.

[63] Pieter Kraaijeveld and Marek J. Druzdzel. Genierate: An interactive generator of diagnostic bayesian network models. In *Working Notes of the 16th International Workshop on Principles of Diagnosis (DX-05)*, pages 175–180, 2005.

[64] Eric S. Lander. *The Human Genome Project.* MIT World, New York, 2002.

[65] Pedro Larrañaga, Mikel Poza, Yosu Yurramendi, Roberto H. Murga, and Cindy M. H. Kuijpers. Structure learning of bayesian networks by genetic algorithms: A performance analysis of control parameters. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(9):912–926, 1996.

[66] Kathryn B. Laskey and Paulo C. G. Da Costa. Of starships and klingons: Bayesian logic for 23rd century. In *Proc. UAI-05*, pages 346–353, 2005.

[67] Steffen L. Lauritzen. *Graphical Models.* Oxford University Press, Oxford, UK, 1996.

[68] Steffen L. Lauritzen and David J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, 50(2):157–224, 1988.

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

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

[71] Makis Leontidis and Constantin Halatsis. Supporting learner's needs with an ontology-based bayesian network. *Advanced Learning Technologies, IEEE International Conference on*, 0:579–583, 2009.

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

[73] Gavin MacBeath and Stuart L. Schreiber. Printing proteins as microarrays for high-throughput function determination. *Science*, 289:1760–1763, 2000.

[74] A. Maedche, B. Motik, L. Stojanovic, and N. Stojanovic. User-driven ontology evolution management. pages 285–300. Springer-Verlag, 2002.

[75] Subramani Mani and Gregory F. Cooper. A study in causal discovery from population-based infant birth and death records. In *Proceedings of the AMIA Annual Fall Symposium 1999, Hanley and Belfus Publishers*, pages 315–319, 1999.

[76] Christopher Meek. Causal inference and causal explanation with background knowledge. In *Proceedings of the Eleventh Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-95)*, pages 403–41, San Francisco, CA, 1995. Morgan Kaufmann.

[77] Stijn Meganck, Philippe Leray, and Bernard Manderick. Learning causal bayesian networks from observations and experiments: A decision theoretic approach. In Vicenç Torra, Yasuo Narukawa, Aïda Valls, and Josep Domingo-Ferrer, editors, *MDAI*, volume 3885 of *Lecture Notes in Computer Science*, pages 58–69. Springer, 2006.

[78] Prasenjit Mitra, Natasha F. Noy, and Anuj R. Jaiswal. Omen: A probabilistic ontology mapping tool. In Yolanda Gil, Enrico Motta, V. Richard Benjamins, and Mark A. Musen, editors, *International Semantic Web Conference*, volume 3729 of *Lecture Notes in Computer Science*, pages 537–547. Springer, 2005.

[79] Kevin P. Murphy. Active learning of causal bayes net structure. Technical report, University of California, Berkeley, USA, 2001.

[80] Richard E. Neapolitan. *Learning Bayesian Networks.* Prentice Hall, 2003.

[81] Natalya F. Noy, Michel Klein, and De Boelelaan A. Ontology evolution: Not the same as schema evolution, 2003.

[82] Martin Oti and Han G. Brunner. The modular nature of genetic diseases. *Clin. Genet.*, 71(1):1–11, 2006.

[83] Zdzislaw Pawlak. Rough sets. *International Journal of Information and Computer Sciences*, 11(5):341–356, 1982.

[84] Judea Pearl. Fusion, propagation, and structuring in belief networks. *Artif. Intell.*, 29(3):241–288, 1986.

[85] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.

[86] Judea Pearl. Jeffrey's rule, passage of experience, and neo-bayesianism. *In H. Kyburg et al., Knowledge Representation and Defeasible Reasoning, Kluwer Academic, Dordrecht*, pages 245–265, 1990.

[87] Judea Pearl. Graphical models, causality and intervention. In *Statistical Science*, volume 8, pages 266–269, 1993.

[88] Judea Pearl. *Causality: Models, reasoning and inference.* Cambridge University Press, Cambridge, MA, 2000.

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

[90] Peter Plessers and Olga De Troyer. Ontology change detection using a version log. In *In Proceeding of the 4th International Semantic Web Conference*, pages 578–592. Springer, 2005.

[91] Daniel J. Povinelli and Derek C. Penn. Causal cognition in human and nonhuman animals: A comparative, critical review. *Annual Review of Psychology*, 58:97–118, 2007.

[92] Roy Rada, Hafedh Mili, Ellen Bicknell, and Maria Blettner. Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man and Cybernetics*, 19(1):17–30, 1989.

[93] Narendran Ramakrishnan and Ananth Y. Grama. Data mining: from serendipity to science. *IEEE Computer*, 32(8):34–37, 1999.

[94] Royston M. Roberts. *Serendipity - Accidental discovery in Science.* Wiley, New York, 1989.

[95] Robert W. Robinson. Counting unlabeled acyclic digraphs. In C. H. C. Little, editor, *Combinatorial Mathematics V*, volume 622 of *Lecture Notes in Mathematics*, pages 28–43, Berlin, 1977. Springer.

[96] David M. Sobel and Jessica A. Sommerville. Rationales in children's causal learning from others' actions. *Cognitive Development*, pages 70–79, 2009.

[97] Paul T. Spellman, Gavin Sherlock, Michael Q. Zhang, Vishwanath R. Iyer, Kirk Anders, Michael B. Eisen, Patrick O. Brown, David Botstein, and Bruce Futcher. Comprehensive identification of cell cycle-regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. *Molecular Biology of the Cell*, 9(12):3273–3297, December 1998.

[98] Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction and Search.* MIT Press, 2001.

[99] Ljiljana Stojanovic, Alexander Maedche, Nenad Stojanovic, and Rudi Studer. Ontology evolution as reconfiguration-design problem solving. In *K-CAP*, pages 162–171, 2003.

[100] Ljiljana Stojanovic, Nenad Stojanovic, and Siegfried Handschuh. Evolution of the metadata in the ontology-based knowledge management systems. In Mirjam Minor and Steffen Staab, editors, *German Workshop on Experience Management*, volume 10 of *LNI*, pages 65–77. GI, 2002.

[101] Michael 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*, 1993.

[102] Jie Tang, Juanzi Li, Bangyong Liang, Xiaotong Huang, Yi Li, and Kehong Wang. Using bayesian decision for ontology mapping. *Journal of Web Semantics*, page 2006, 2006.

[103] Jin Tian and Ilya Shpitser. *On Identifying Causal Effects.* In R. Dechter, H. Geffner, and J. Halpern (Eds.), Heuristics, Probability and Causality: A Tribute to Judea Pearl, College Publications, 2010.

[104] Simon Tong and Daphne Koller. Active learning for structure in bayesian networks. In *In International Joint Conference on Artificial Intelligence*, pages 863–869, 2001.

[105] Ioannis Tsamardinos, Laura Brown, and Constantin Aliferis. The max-min hill-climbing bayesian network structure learning algorithm. *Machine Learning*, 65(1):31–78, October 2006.

[106] Linda C. Van Der Gaag and John-Jules Ch. Meyer. Informational independence: Models and normal forms. Technical Report UU-CS-1997-17, Department of Information and Computing Sciences, Utrecht University, 1997.

[107] James Woodward. *Making things happen: A theory of causal explanation.* Oxford University Press, Oxford, 2003.

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

[109] Yi Yang and Jacques Calmet. Ontobayes: An ontology-driven uncertainty model. *International Conference on Computational Intelligence for Modelling, Control and Automation (CIMCA)*, pages 457–463, 2005.

[110] Lotfi A. Zadeh. Fuzzy sets. *Information Control*, 8:338–353, 1965.

[111] Yuan Cao Zhang, Diarmuid Ó Séaghdha, Daniele Quercia, and Tamas Jambor. Auralist: introducing serendipity into music recommendation. In Eytan Adar, Jaime Teevan, Eugene Agichtein, and Yoelle Maarek, editors, *WSDM*, pages 13–22. ACM, 2012.

Appendix A

The OWL (Web Ontology Language) is a newly recommended semantic language for web resources of W3C (World Wide Web Consortium). The purpose of this language is to present information by categories of the objects and their interrelationships. As shown on Figure 6.1, OWL extends and supports earlier W3C standard, such as XML, XML Schema, RDF and RDF Schema, providing richer vocabulary and modeling primitives.

The main concepts available in OWL are:

- **Class**: A class defines a group of individuals that belong together because they share some properties;
- **rdfs:subClassOf**: Class hierarchies may be created by making one or more statements that a class is a subclass of another class;
- **rdf:Property**: Properties can be used to state relationships between individuals or from individuals to data values;

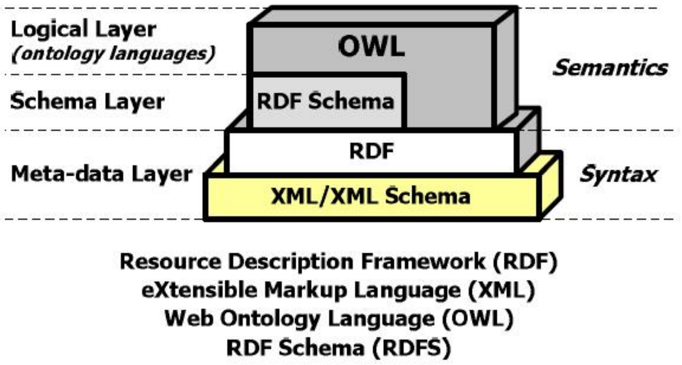


Figure 6.1: OWL in the semantic web architecture

- **rdfs:subPropertyOf**: Property hierarchies may be created by making one or more statements that a property is a subproperty of one or more other properties;
- **rdfs:domain**: A domain of a property limits the individuals to which the property can be applied;
- **rdfs:range**: The range of a property limits the individuals that the property may have as its value; and
- **Individual**: Individuals are instances of classes, and properties may be used to relate one individual to another.

OWL development together with many tools for ontology construction (Protégé [1], OntoStudio [2], etc) made ontologies quite widespread and the number of available ontologies is fastly growing. OWL provides three increasingly expressive sub-languages designed for use by specific communities of implementers and users:

- a) OWL Lite (is least expressive, suitable for simple class hierarchy and simple constraints and useful for quick migration path for thesauri and other taxonomies),
- b) OWL DL (is more expressive, retains Computational Completeness that is, all conclusions are guaranteed to be computable and has Decidability that is, all computations will finish in finite time, and is based on Description Logic),
- c) OWL Full (is most expressive and has syntactically freedom of RDF and has no computational guarantees but allows an ontology to augment the meaning of the pre-defined (RDF or OWL) vocabulary and is not suitable for auto-reasoning).

Simultaneously, the set of OWL ontologies represent a knowledge base for OWL reasoner. A reasoner can be defined as a system that allows the

[1] http://protege.stanford.edu/

[2] http://www.ontoprise.de/en/home/products/ontostudio/

inference of implicitly knowledge from the knowledge that is explicitly contained in a knowledge base. There are a wide range of OWL reasoners in modern knowledge-based systems. Each version has its own functional and non-functional trade-offs including computational complexity, semantic expressiveness, and processor load. There are a number of semantic reasoners such as: Pellet [3], RacerPro [4] and Fact++ [5].

[3] http://clarkparsia.com/pellet/

[4] http://www.racer-systems.com/

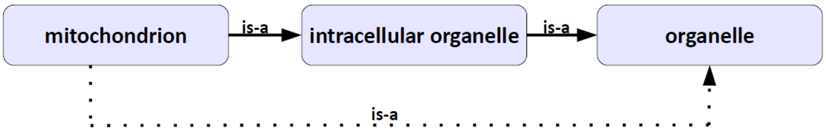
[5] http://code.google.com/p/factplusplus/

Appendix B

Inference is an important aspect of ontology driven applications which has been repeatedly mentioned in the previous chapters when describing the SemCaDo approach. In what follows we give the basic reasoning rules of the GO relations:

- *Reasoning over is-a:*

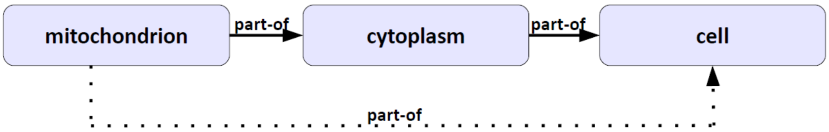
    - is-a $\circ$ is-a $\sqsubseteq$ is-a



    The is-a relation is transitive, which means that if A is a B, and B is a C, then we can infer that A is a C.

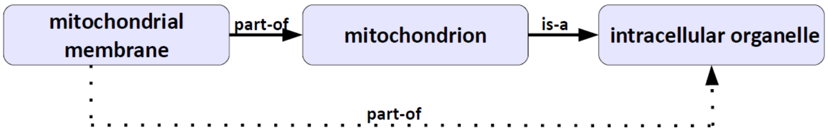
- *Reasoning over part-of:*

    - part-of $\circ$ part-of $\sqsubseteq$ part-of



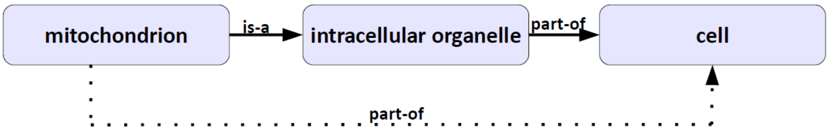
    Like is-a, part-of is transitive: if A is part-of B, and B is part-of C then A is part-of C

    - part-of $\circ$ is-a $\sqsubseteq$ part-of



If a part of relation is followed by an is a relation, it is equivalent to a part of relation; if A is part of B, and B is a C, we can infer that A is part of C.

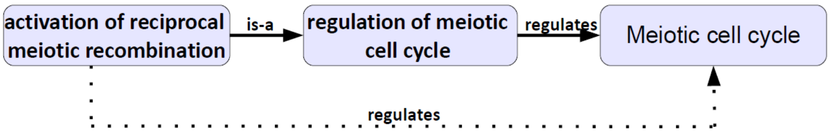
- is-a ∘ part-of ⊑ part-of



If the order of the relationships is reversed, the result is the same; if A is a B, and B is part of C, A is part of C.

- *Reasoning over regulates*:

- is-a ∘ regulates ⊑ regulates

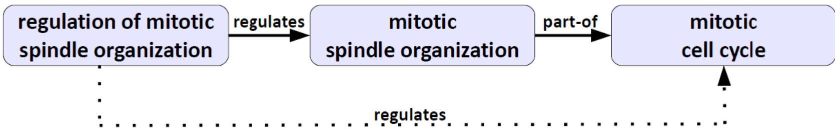

activation of reciprocal meiotic recombination
is-a
regulation of meiotic cell cycle
regulates
Meiotic cell cycle
regulates

If A is a B, and B regulates C, we can infer that A regulates C. This rule is true for positively regulates and negatively regulates.

- regulates ∘ is-a ⊑ regulates

If we switch the relations around, so that A is a B, and B regulates C, we can again infer that A regulates C. This rule also holds true for the positively regulates and negatively regulates relations.

- regulates ∘ part-of ⊑ regulates


regulation of mitotic spindle organization
regulates
mitotic spindle organization
part-of
mitotic cell cycle
regulates

The GO also uses the rule that if B is part of C, any A that regulates B also regulates C.

Appendix C

SEMCADO implementation tricks

This Annexe gives additional guidance and provides hands on useful implementation tricks.

C.1) Divide & conquer: work on connected non directed components

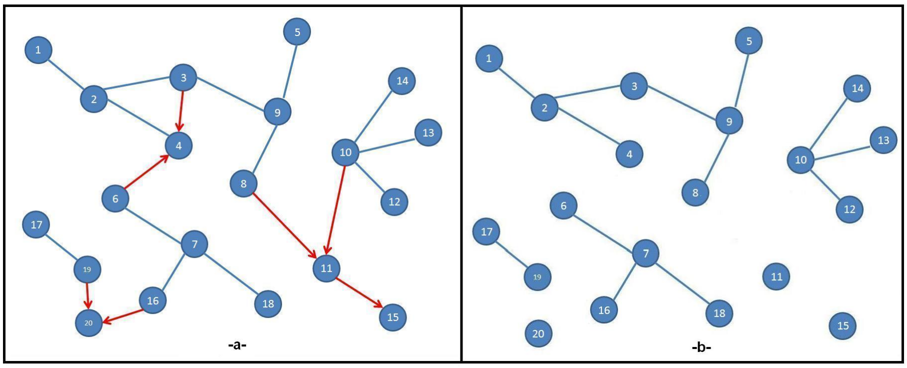


Figure 6.2: Graph decomposition

The major problem when computing utilities is that we need to know the exact number of class equivalence elements. In order to avoid this problem, we will adopt a graph decomposition strategy, which consists on eliminating directed edges from the studied PDAG.

We opt for this solution since the instantiation of each undirected subgraph is totally independent of other substructures. This forms the basis for a divide-and-conquer method working on reduced graphs (i.e. undirected components) and reducing temporal complexity.

Example 6.1. *Figure 6.2 shows an example of graph decomposition. In the initial graph, we have exactly 13 non-directed edges (edges in blue color). This imply that we will have* 2^{13} *or 8192 equivalent class members to take into consideration while calculating SEMCADO utilities. While deleting directed edges (red color), we pass from only one graph to four reduced components without counting single node components. In term of equivalent class members, we will obtain:* $2^6 + 2^3 + 2^3 + 2$*=64+8+8+2=82*

As the learning process proceeds, we obtain more and more small-scale substructures.

C.2) **Using prior restrictions within independence-based learning algorithms**

The learning algorithms based on independence tests typically start from a complete, undirected graph and delete recursively edges based on conditional independence decisions given some subset of nodes. Then they have to direct edges to form head-to-head patterns or v-structures (triplets of nodes x, y, z such that x and y are not adjacent and the arcs x → z and y → z exist).

Both activities are guided by the results of χ^2 independence test applied to the available data. This yields an undirected graph which can then be partially directed and further extended to represent the underlying DAG.

For instance, when using PC algorithm [98], we first eliminate as many edges as we can, and after we give direction to some of the non-removed edges by forming v-structures. Finally, several additional edges may be directed by using PC rules.

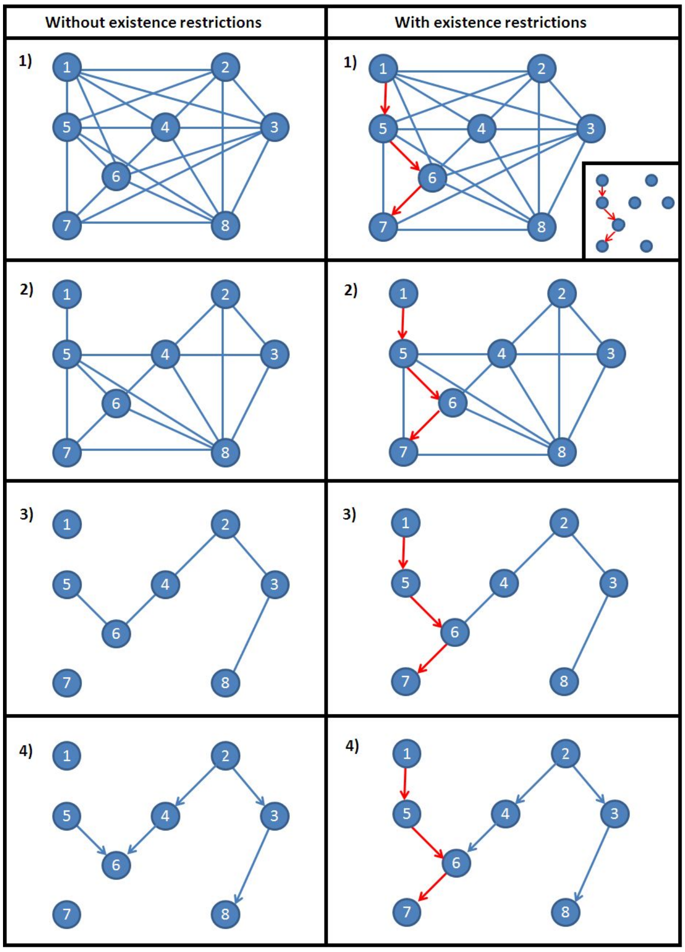


Figure 6.3: Comparison between using PC algorithm without (resp. with) prior restrictions

In our context, a simple method to integrate the set of presence restrictions is to fix them from the beginning in the complete undirected graph and proceed the independence test calculations. In figure 6.3, we show the different PC iterations when proceeding without (resp. with) prior restrictions.

The principle advantage when integrating presence restrictions is to reduce the size of the sets of nodes which are candidate to form the separating sets employed by the χ^2 independence tests. This technique reduces considerably the exponential time needed to return the representative structure.

C.3) **Principle programs**

Algorithm 2 Node Experimentation

```
Require: $Original\_CBN, Actual\_BN, Node\_exp, Obs.DataSet$
Neigh_List ← Find_Non_Directed_Neighbors(Actual_BN, Node_exp)
num_neighbors ← Neigh_List.size()
Exp.DataSet ← Generate_Exp_DataSet(Actual_BN, Node_exp, nbSamples, FileName)
Chi2Result ← Chi2_Adequation_Test(Obs.DataSet, Exp.DataSet)
if Chi2Result=false then

  for $i = 0$ to num_neighbors do
    Add_edge(Node_exp, Neigh_List[i], Actual_BN)
  end for
else

  for $i = 0$ to num_neighbors do
    Add_edge(Neigh_List[i], Node_exp, Actual_BN)
  end for
end if
return $Actual\_BN$
```

Algorithm 3 Maximax-MyCaDo

```
Require: Actual_BN, Node_X, Exp_Cost, Obs_Cost
nb_instantiations ← 0
nb_Poss_Struct_inst ← 0
Poss_Struct ← 0
All_Obs_Cost ← Obs_Cost[Node_X]
nbNodes ← getVariables(Actual_BN)
Neigh_List ← Find_Non_Directed_Neighbors(Actual_BN, Node_X)

for $i = 0$ to Neigh_List.size() do
    nb_instantiations ← nb_instantiations × 2
    All_Obs_Cost ← Obs_Cost[Neigh_List[i]]
end for
Mat_Inst ← create_mat_instantiations(Neigh_List)

for $i = 0$ to nb_instantiations do
    Poss_Struct_inst ← 0
    Resulting_Network ← Add_edges_instantiation(Mat_Inst[i])
    Non_Dir_EdgeList_before ← Get_Non_Directed_EdgeList(Resulting_Network)
    Resulting_Network ← Apply_PC_Rules(Resulting_Network)
    Non_Dir_EdgeList_after ← Get_Non_Directed_EdgeList(Resulting_Network)
    Mat_Poss_Struct ← create_mat_instantiations(Non_Dir_EdgeList_before)

    for $i = 0$ to Non_Dir_EdgeList_before.size() do
        nb_Poss_Struct_inst ← nb_Poss_Struct_inst × 2
    end for

    for $i = 0$ to nb_Poss_Struct_inst do
        Resulting_Network ← Edge_Inf_Result(Actual_BN, Mat_Poss_Struct)
        Test_Verif ← V_structure_Test(Resulting_Network)
        if Test_Verif=false then

            Poss_Struct_inst ← Poss_Struct_inst+1
            Poss_Struct ← Poss_Struct+1
        end if
    end for
    Inferred ← Non_Dir_EdgeList_after - Non_Dir_EdgeList_before
    Inferred_List ← Inferred
end for
Utility ← (Neigh_List.size() + Max (Inferred_List)) / (Exp_Cost +
```

Algorithm 4 Maximax-SemCaDo

```
Require: $Actual\_BN, Node\_X, Exp\_Cost, Obs\_Cost$, Ontology
Neigh_List ← Find_Non_Directed_Neighbors(Actual_BN, Node_X)
Neigh_List ← Node_X
All_Subsumers ← Identify_Direct_Subsumers(Neigh_List, Ontology)
MSCS ← Identify_MSCS(All_Subsumers, Ontology)
Sem_Inertia ← Get_Semantical_Inertia(Neigh_List, MSCS, Ontology)

nb_instantiations ← 0
nb_Poss_Struct_inst ← 0
Poss_Struct ← 0
All_Obs_Cost ← Obs_Cost[Node_X]
nbNodes ← getVariables(Actual_BN)
Neigh_List ← Find_Non_Directed_Neighbors(Actual_BN, Node_X)

for $i = 0$ to Neigh_List.size() do
    nb_instantiations ← nb_instantiations × 2
    All_Obs_Cost ← Obs_Cost[Neigh_List[i]]
end for
Mat_Inst ← create_mat_instantiations(Neigh_List)

for $i = 0$ to nb_instantiations do
    { MaxiMax-MyCaDo instructions from 15 to 36 }
    Inf_Nodes ← Get_Nodes(Inferred_List)
    Inf_All_Subsumers ← Identify_Direct_Subsumers(Inf_Nodes, Ontology)
    Inf_MSCS ← Identify_MSCS(Inf_All_Subsumers, Ontology)
    Inf_Gain ← Get_Semantical_Inertia(Inf_Nodes, Inf_MSCS, Ontology)
end for
Utility ← (Sem_Inertia + Max (Inf_Gain)) / (Exp_Cost + Obs_Cost)
return Utility
```

Algorithm 5 Maximin-MyCaDo

```
Require: $Actual\_BN, Node\_X, Exp\_Cost, Obs\_Cost$
nb_instantiations ← 0
nb_Poss_Struct_inst ← 0
Poss_Struct ← 0
All_Obs_Cost ← Obs_Cost[Node_X]
nbNodes ← getVariables(Actual_BN)
Neigh_List ← Find_Non_Directed_Neighbors(Actual_BN, Node_X)

for $i = 0$ to Neigh_List.size() do
  nb_instantiations ← nb_instantiations × 2
  All_Obs_Cost ← Obs_Cost[Neigh_List[i]]
end for
Mat_Inst ← create_mat_instantiations(Neigh_List)

for $i = 0$ to nb_instantiations do
  Poss_Struct_inst ← 0
  Resulting_Network ← Add_edges_instantiation(Mat_Inst[i])
  Non_Dir_EdgeList_before ← Get_Non_Directed_EdgeList(Resulting_Network)
  Resulting_Network ← Apply_PC_Rules(Resulting_Network)
  Non_Dir_EdgeList_after ← Get_Non_Directed_EdgeList(Resulting_Network)
  Mat_Poss_Struct ← create_mat_instantiations(Non_Dir_EdgeList_before)

  for $i = 0$ to Non_Dir_EdgeList_before.size() do
    nb_Poss_Struct_inst ← nb_Poss_Struct_inst × 2
  end for

  for $i = 0$ to nb_Poss_Struct_inst do
    Resulting_Network ← Edge_Inf_Result(Actual_BN,
    Mat_Poss_Struct)
    Test_Verif ← V_structure_Test(Resulting_Network)
    if Test_Verif=false then

      Poss_Struct_inst ← Poss_Struct_inst+1
      Poss_Struct ← Poss_Struct+1
    end if
  end for
  Inferred ← Non_Dir_EdgeList_after - Non_Dir_EdgeList_before
  Inferred_List ← Inferred
end for
Utility ← (Neigh_List.size() + Min (Inferred_List)) / (Exp_Cost +
```

Algorithm 6 Maximin-SemCaDo

```
Require: $Actual\_BN, Node\_X, Exp\_Cost, Obs\_Cost$, Ontology
Neigh_List ← Find_Non_Directed_Neighbors(Actual_BN, Node_X)
Neigh_List ← Node_X
All_Subsumers ← Identify_Direct_Subsumers(Neigh_List, Ontology)
MSCS ← Identify_MSCS(All_Subsumers, Ontology)
Sem_Inertia ← Get_Semantical_Inertia(Neigh_List, MSCS, Ontology)

nb_instantiations ← 0
nb_Poss_Struct_inst ← 0
Poss_Struct ← 0
All_Obs_Cost ← Obs_Cost[Node_X]
nbNodes ← getVariables(Actual_BN)
Neigh_List ← Find_Non_Directed_Neighbors(Actual_BN, Node_X)

for $i = 0$ to Neigh_List.size() do
    nb_instantiations ← nb_instantiations × 2
    All_Obs_Cost ← Obs_Cost[Neigh_List[i]]
end for
Mat_Inst ← create_mat_instantiations(Neigh_List)

for $i = 0$ to nb_instantiations do
    { MaxiMin-MyCaDo instructions from 15 to 36 }
    Inf_Nodes ← Get_Nodes(Inferred_List)
    Inf_All_Subsumers ← Identify_Direct_Subsumers(Inf_Nodes, Ontology)
    Inf_MSCS ← Identify_MSCS(Inf_All_Subsumers, Ontology)
    Inf_Gain ← Get_Semantical_Inertia(Inf_Nodes, Inf_MSCS, Ontology)
end for
Utility ← (Sem_Inertia + Min (Inf_Gain)) / (Exp_Cost + Obs_Cost)
return Utility
```

Algorithm 7 Expected-Utility-MyCaDo

```
Require: $Actual\_BN, Node\_X, Exp\_Cost, Obs\_Cost$
nb_instantiations ← 0
nb_Poss_Struct_inst ← 0
Poss_Struct ← 0
All_Obs_Cost ← Obs_Cost[Node_X]
nbNodes ← getVariables(Actual_BN)
Neigh_List ← Find_Non_Directed_Neighbors(Actual_BN, Node_X)

for $i = 0$ to Neigh_List.size() do
  nb_instantiations ← nb_instantiations × 2
  All_Obs_Cost ← Obs_Cost[Neigh_List[i]]
end for
Mat_Inst ← create_mat_instantiations(Neigh_List)

for $i = 0$ to nb_instantiations do
  Poss_Struct_inst ← 0
  Resulting_Network ← Add_edges_instantiation(Mat_Inst[i])
  Non_Dir_EdgeList_before ← Get_Non_Directed_EdgeList(Resulting_Network)
  Resulting_Network ← Apply_PC_Rules(Resulting_Network)
  Non_Dir_EdgeList_after ← Get_Non_Directed_EdgeList(Resulting_Network)
  Mat_Poss_Struct ← create_mat_instantiations(Non_Dir_EdgeList_before)

  for $i = 0$ to Non_Dir_EdgeList_before.size() do
    nb_Poss_Struct_inst ← nb_Poss_Struct_inst × 2
  end for

  for $i = 0$ to nb_Poss_Struct_inst do
    Resulting_Network ← Edge_Inf_Result(Actual_BN, Mat_Poss_Struct)
    Test_Verif ← V_structure_Test(Resulting_Network)
    if Test_Verif=false then

      Poss_Struct_inst ← Poss_Struct_inst+1
      Poss_Struct ← Poss_Struct+1
    end if
  end for
  Inferred ← Non_Dir_EdgeList_after - Non_Dir_EdgeList_before
  Inferred_List ← Inferred
end for
Inf_Inst←Inf_Inst+ (Inferred × Poss_Struct_inst/Poss_Struct)
```

Algorithm 8 Expected-Utility-SemCaDo

```
Require: $Actual\_BN, Node\_X, Exp\_Cost, Obs\_Cost$, Ontology
Neigh_List ← Find_Non_Directed_Neighbors(Actual_BN, Node_X)
Neigh_List ← Node_X
All_Subsumers ← Identify_Direct_Subsumers(Neigh_List, Ontology)
MSCS ← Identify_MSCS(All_Subsumers, Ontology)
Sem_Inertia ← Get_Semantical_Inertia(Neigh_List, MSCS, Ontology)

nb_instantiations ← 0
nb_Poss_Struct_inst ← 0
Poss_Struct ← 0
All_Obs_Cost ← Obs_Cost[Node_X]
nbNodes ← getVariables(Actual_BN)
Neigh_List ← Find_Non_Directed_Neighbors(Actual_BN, Node_X)

for $i = 0$ to Neigh_List.size() do
    nb_instantiations ← nb_instantiations × 2
    All_Obs_Cost ← Obs_Cost[Neigh_List[i]]
end for
Mat_Inst ← create_mat_instantiations(Neigh_List)

for $i = 0$ to nb_instantiations do
    { Expected-Utility-MyCaDo instructions from 15 to 36 }
    Inf_Nodes ← Get_Nodes(Inferred_List)
    Inf_All_Subsumers ← Identify_Direct_Subsumers(Inf_Nodes, Ontology)
    Inf_MSCS ← Identify_MSCS(Inf_All_Subsumers, Ontology)
    Inf_Gain ← Get_Semantical_Inertia(Inf_Nodes, Inf_MSCS, Ontology)
end for
Inf_Inst←Inf_Inst+ (Inf_Gain × Poss_Struct_inst/Poss_Struct)
Utility ← (Sem_Inertia + Inf_Inst) / (Exp_Cost + Obs_Cost)
return Utility
```

Algorithm 9 Order-Edges

```
Require: DAG,
Topological_Order(DAG)
i ← 0
A ← Unordered_Edges(DAG)
while A $\neq$ 0 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 while
return DAG with labeled total order on edges
```

Algorithm 10 DAG-to-CPDAG

```
Require: Actual-BN,
 1: Order-Edges(Actual-BN)
 2: Label every edge in Actual-BN as "unknown"
 3: while there are edges labeled "unknown" in Actual-BN do
 4:   Lat x → y be the lowest ordered edge that is labeled "unknown"
 5:
 6:   for every edge w → x labeled "compelled" do
 7:     if w is not a parent of y then
 8:       Label x → y and every edge incident into y with "compelled"
 9:       GoTo 3
10:     else
11:       Label w → y with "compelled"
12:     end if
13:   end for
14:   if there exists and edge z → y such that $z \neq x$ and z is not a parent
      of x then
15:     Label x → y and all "unknown" edges incident into y with "compelled"
16:   else
17:     Label x → y and all "unknown" edges incident into y with "reversible"
18:   end if
19: end while
20: return DAG with each edge labeled either "compelled" or "reversible"
```