UNIVERSITÉ DE TUNIS
INSTITUT SUPÉRIEUR DE GESTION DE TUNIS

MÉMOIRE DE MASTÈRE
EN INFORMATIQUE ET GESTION DE LA CONNAISSANCE

Learning possibilistic networks from possibilistic datasets

Par
Maroua HADDAD

Directeur : Nahla Ben Amor,Maître de Conférence,
ISG-Université de Tunis.

Année Universitaire 2011/2012

Contents

List of Figures

List of Tables

Introduction

Graphical models are efficient and compact knowledge representation and reasoning tools. That's why the interest in graphical models grew rapidly and is still growing to this day.

Possibility theory [10, 22] offers an appropriate framework to handle preferences, imprecision and uncertainty which occur in many real-life problems.

In our work, we are interested in possibilistic networks which are compact representations of possibility distributions. Their popularity is due to its capacity to model uncertainty, imprecision and incompleteness characterizing information in simple and intuitive way. They represent the counterpart of Bayesian networks [17, 20] in the possibilistic framework.

Learning possibilistic networks is a new promising area of research because constructing them by human experts is tedious and time consuming. This work proposes a new approach to learn possibilistic networks from possibilistic datasets (containing attributes described by possibility distributions).

In this work, we will focus on the problem of how we can estimate marginal possibility distributions from datasets and how we can apply it to learn possibilistic networks structure.

This report is organized as follows: Chapter 1 presents theoretical aspects regarding possibility theory and several similarity measures. It details how we can compare possibility distributions. Chapter 2 presents possibilistic networks and cited methods existing in literature for learning them from data. Finally, in chapter 3, we will propose a new method to learn possibilis-

tic networks from possibilistic datasets. We will detail how we can induce possibility distributions using the notion of similarity measures.

Chapter 1

Basics of possibility theory and possibilistic similarity measures

1.1 Introduction

Possibility theory was coined by Zadeh in [22] as an extension of fuzzy sets theory and Didier Dubois and Henri Prade contributed to its development [10]. The success of possibility theory is due to its capacity to handle uncertainty and imprecision in simple way and to provide a graded semantics to natural language statements.

In several situations, we need to compare uncertain and imprecise information. So, the notion of similarity is very important in almost every scientific field. That's why, many works have proposed similarity measures that can be applied in the possibilistic case.

This chapter provides an overview of possibility theory and details the notion of similarity. It is organized as follows: In section 1.1, we introduce basics of possibility theory. In section 1.2, we list some possibilistic similarity measures and their properties that will be used later in Chapter 3.

1.2 Basics of possibility theory

Possibility theory is a non-classical theory of uncertainty devoted to handle imperfect information. This section gives some basic elements of this theory. We first explain the notion of a possibility distribution. Then, we introduce two dual measures: possibility and necessity. Finally, we define the notion of conditioning in the possibilistic case: qualitative and quantitative settings.

1.2.1 Possibility distribution

The basic building block in possibility theory is the notion of possibility distribution. It is a representation of knowledge concerning a state of the world. It corresponds to a mapping from the universe of discourse $\Omega = \{\omega_1, \omega_2, ..., \omega_n\}$ to the scale L = [0,1] encoding our knowledge on the real world states.

$$\pi : \Omega \rightarrow L$$
$$\omega \rightarrow \pi_x(\omega)$$

It reflects the possibility degree that $x = \omega$ is true.

- $\pi_x(\omega) = 0$ if $x = \omega$ is rejected as impossible.
- $\pi_x(\omega) = 1$ if $x = \omega$ is fully possible.
- We are in the case of complete knowledge if:

$$\exists \omega_i \in \Omega / \pi(\omega_i) = 1 \; and \; \forall \omega \in \Omega - \{\omega_i\}, \pi(\omega) = 0. \tag{1.1}$$

- We are in the case of total ignorance if:

$$\forall \omega \in \Omega, \pi(\omega) = 1. \tag{1.2}$$

- A possibility distribution π is normalized if it contains at least one state $\omega \in \Omega$ that is fully possible. This property can be defined as:

$$\max_{\omega} \pi(\omega) = 1 \tag{1.3}$$

- The degree of inconsistency, denoted by Inc, is defined as follows:

$$Inc(\pi) = 1 - \max_{\omega \in \Omega}\{\pi(\omega)\} \tag{1.4}$$

The scale L has two interpretations:

- Ordinal interpretation : The possibility distribution is a mapping from the universe of discourse Ω to an ordinal scale where only the order of values is important. In fact, The scale encodes an ordering between different degrees.

- Numerical interpretation : The possibility distribution is a mapping from the universe of discourse Ω to a numerical scale where values have sense. We are interested with real numbers represented by degrees of possibility that can be manipulated by arithmetic operators.

In what follows, we present some basic elements in possibility theory that we need in chapter 3.

- A possibility distribution π is more specific than π' if:

$$\forall \omega_i \in \Omega, \pi(\omega_i) \leq \pi'(\omega_i) \tag{1.5}$$

- U-uncertainty is the measure that assess the amount of imprecision in a given possibility distribution. This measure, denoted by U, is defined as follows:

  Let $\pi = \{\pi_{(1)}, \pi_{(2)}, ..., \pi_{(n)}\}$ be an ordered distribution (descending order)

$$U(\pi) = [\sum_{i=1}^{n}(\pi_{(i)} - \pi_{(i+1)})\ log_2\ i] + (1 - \pi_{(1)})\ log_2\ n \tag{1.6}$$

  where $\pi_{n+1} = 0$ by convention.

1.2.2 Possibility and necessity measures

Possibility and necessity measures named Π and N are tow dual measures used in the possibilistic case.

Given a possibility distribution π, we can define for any subset $A \in \Omega$, $\Pi(A)$ and $N(A)$ as follows:

$$\Pi(A) = \max_{\omega \in A} \pi_x(\omega) \tag{1.7}$$

Π assess at what level A is consistent with the knowledge represented by π.

$$N(A) = 1 - \Pi(\bar{A}) = \min_{w \notin A}(1 - \pi_x(\omega)) \tag{1.8}$$

N assess at what level $\neg A$ is impossible.

1.2.3 Possibilistic conditioning

The possibilistic conditioning is crucial notion that consists in revising our initial knowledge, represented by a possibility distribution π by the arrival of a new absolutely certain piece of information.

In the following, we discuss two types of possibilistic conditioning depending on the interpretation of the scale used : ordinal or numerical.

- In an ordinal setting, we assign to best elements of A, the maximal possibility degree (i.e.1), then we obtain:

$$\pi(\omega \mid_m A) = \begin{cases} 1 & if\ \pi(\omega) = \Pi(A) \text{ and } \omega \in A \\ \pi(\omega) & if\ \pi(\omega) < \Pi(A) \text{ and } \omega \in A \\ 0 & otherwise. \end{cases} \tag{1.9}$$

  This corresponds to *min-based* conditioning.

- In a numerical setting (if the definition makes sense in the ranking scale), we proportionally shift up all elements of A:

$$\pi(\omega \mid_p A) = \begin{cases} \frac{\pi(\omega)}{\Pi(A)} & \text{if } \omega \in A \\ 0 & \textit{otherwise.} \end{cases} \tag{1.10}$$

  This corresponds to *product-based* conditioning.

1.3 Possibilistic similarity measures

The mathematical concept of similarity measures is fundamentally important in almost every scientific field. It is used essentially to compare two objects and it reflects the degree of closeness between them. This concept is useful in data mining, information retrieval, pattern recognition, natural and social science, etc.

In the possibilistic case, few works have developed or adapted similarity measures. They are used to compare two objects described by possibility distributions on the same universe of discourse.

In this section, the first part will be devoted to enumerate properties that should satisfy a similarity measure. In the second part, we list several similarity measures.

1.3.1 Properties of a possibilistic similarity measure

[14, 16] propose several basic natural properties of similarity measure (1.1-1.6) and also some extended ones (1.7-1.10).

Property 1.1. ***Non negativity:*** *The similarity between two possibility distributions must be positive:* $s(\pi_1, \pi_2) \geq 0$.

Property 1.2. ***Symmetry****: The order of possibility distributions does not influence the similarity value:* $s(\pi_1, \pi_2) = s(\pi_2, \pi_1)$.

Property 1.3. ***Upper bound and non-degeneracy:*** *The similarity between identical distributions is equal to 1. So, the upper bound of similarity is equal to 1 too:* $s(\pi_1, \pi_2) \leq 1$.

Property 1.4. ***Lower bound :*** *The similarity between two contradictory possibility distributions is equal to 0. So, the lower bound of similarity is equal to 0 too:* $s(\pi_1, \pi_2) \geq 0$.

Example 1.1. *Let us consider the two distributions* $\pi_1 = \{0, 1, 0\}$ *and* $\pi_2 = \{1, 0, 1\}$*:* $s(\pi_1, \pi_2) = 0$.

Property 1.5. ***Large inclusion :*** *If we have three possibility distributions and the first one is more specific than the second which is in turn more specific than the third and they agree for at least one state, the similarity between first and second distributions is greater or equal to the similarity between the second and the third ones.*
Let π_1, π_2 *and* π_3 *be three possibility distributions satisfying* $Inc(\pi_1, \pi_2) = Inc(\pi_1, \pi_3) = Inc(\pi_2, \pi_3) = 0$ *and* $\forall \omega_i \in \Omega\ \pi_1(\omega_i) \leq \pi_2(\omega_i)$ *and* $\pi_2(\omega_i) \leq \pi_3(\omega_i)$ *then* $s(\pi_1, \pi_2) \geq s(\pi_1, \pi_3)$.

Property 1.6. ***Permutation :*** *The permutation of elements order of possibility distributions does not influence the value of similarity. We denote* π_{p1} *and* π_{p2} *resulting distributions of permutation of indexes.* $s(\pi_1, \pi_2) = s(\pi_{p1}, \pi_{p2})$.

Example 1.2. *Let us consider two possibility distributions* $\pi_1 = \{1, 0.2, 0.3\}$ *and* $\pi_2 = \{0, 1, 0.5\}$*. Let us permute elements, we obtain* $\pi_{p1} = \{0.3, 0.2, 1\}$ *and* $\pi_{p2} = \{0.5, 1, 0\}$*.* $s(\pi_1, \pi_2) = (\pi_{p1}, \pi_{p_2})$.

Property 1.7. ***Strict inclusion :*** *this property is an extension of property(1.5) but another condition should be added: If the possibility distributions are different, the similarity between the first and second distributions is strictly greater to the similarity between the second and the third ones.*
$\forall \pi_1$, π_2 *and* π_3 *s.t.* $\pi_1 \neq \pi_2 \neq \pi_3$ *, if* $\pi_1 \leq \pi_2 \leq \pi_3$ *then* $s(\pi_1, \pi_2) > s(\pi_1, \pi_3)$.

Property 1.8. ***Reaching coherence :*** *If we enhance the degree of a state of two conflicting possibility distributions, we enhance the similarity degree*

between them. Let π_1 *and* π_2 *be two possibility distributions,* $\omega_i \in \Omega$. *Let* π'_1 *and* π'_2 *s.t.* $\forall j \neq i, \pi'_1(\omega) = \pi_1(\omega)$ *and* $\pi'_2(\omega) = \pi_2(\omega)$. *Let* α *s.t.* $\alpha \leq 1 - \max(\pi_1(\omega_i), \pi_2(\omega_i))$, *if* $\pi'_1(\omega_i) = \pi_1(\omega_i) + \alpha$ *and* $\pi'_2(\omega_i) = \pi_2(\omega_i) + \alpha$:

- *if* $Inc(\pi_1 \wedge \pi_2) = Inc(\pi'_1 \wedge \pi'_2)$ *then* $s(\pi_1, \pi_2) = s(\pi'_1, \pi'_2)$
- *if* $Inc(\pi_1 \wedge \pi_2) > Inc(\pi'_1 \wedge \pi'_2)$ *then* $s(\pi_1, \pi_2) < s(\pi'_1, \pi'_2)$

Example 1.3. *Let us consider the two distributions* $\pi_1 = \{1, 0.2, 0.3\}$ *and* $\pi_2 = \{0, 1, 0.5\}$.. *We increase the degree of the third element, we obtain* $\pi_1 = \{1, 0.2, 0.8\}$ *and* $\pi_2 = \{0, 1, 1\}$: $s(\pi_1, \pi'_1) = s(\pi_2, \pi'_2)$.

Property 1.9. ***Mutual convergence :*** *If we have two possibility distributions which disagree in one state and we replace this degree in one of distributions with a closer value, the similarity between possibility distributions increases. Let* π_1 *and* π_2 *be two possibility distributions,* $\exists \omega_i / \pi_1(\omega_i) > \pi_2(\omega_i)$. *let* π'_2 *s.t.* $\pi'_2(\omega_i) \in]\pi_2(\omega_i), \pi_1(\omega_i)]$ *and* $\forall j \neq i, \pi'_2(\omega_j) = \pi_2(\omega_j) : s(\pi_1, \pi'_2) > s(\pi_1, \pi_2)$.

Property 1.10. ***Indifference preserving :*** *If we make the same variation to two distinct states in two possibility distributions, the similarity does not change. Let* π_1 *and* π_2 *be two possibility distributions. Let* π'_1 *and* π'_2 *s.t.*

- $\forall j \neq p, \pi'_1(\omega_j) = \pi_1(\omega_j)$ *and* $\pi'_1(\omega_p) = \pi_1(\omega_p) + \alpha (resp. - \alpha)$.
- $\forall j \neq q, \pi'_2(\omega_j) = \pi_2(\omega_j)$ *and* $\pi'_2(\omega_q) = \pi_2(\omega_q) + \alpha (resp. - \alpha)$.

$$s(\pi_1, \pi'_1) = s(\pi_2, \pi'_2).$$

Example 1.4. *Let us consider the two distributions* $\pi_1 = \{1, 0.2, 0.3\}$ *and* $\pi_2 = \{0, 1, 0.5\}$.. *We increase the degrees of two distinct states, we obtain* $\pi_1 = \{1, 0.7, 0.3\}$ *and* $\pi_2 = \{0, 1, 1\}$: $s(\pi_1, \pi'_1) = s(\pi_2, \pi'_2)$.

1.3.2 Possibilistic similarity and distance measures

In the case of possibility theory, few works were devoted to similarity and distance measures despite their importance in several areas. They are useful to compare two possibility distributions and provide the degree of closeness or difference between them.

In the following, we first list principle similarity and distance measures then we illustrate them by an example.

1.3.2.1 Information closeness

The information closeness measure is one of first works devoted to measure similarity between two possibility distributions. It was introduced in [13]. The underlying idea of information closeness is U-uncertainty concept.

In the following, we denote G(π_1,π_2) the value of information closeness between two distributions π_1, π_2. $\vee$ refers to the maximum operator and $U(\pi)$ is given by equation(1.6).

$$G(\pi_1, \pi_2) = g(\pi_1, \pi_1 \vee \pi_2) + g(\pi_2, \pi_1 \vee \pi_2) \tag{1.11}$$

where $g(\pi_1, \pi_2) = U(\pi_2) - U(\pi_1)$. Thus, the information closeness can also be written as:

$$G(\pi_1, \pi_2) = 2 * U(\pi_1 \vee \pi_2) - U(\pi_1) - U(\pi_2). \tag{1.12}$$

The information closeness measure behaves as a distance but it can be transformed to assess the similarity between two possibility distributions. This derived similarity measure, denoted by S_G, is defined as:

$$S_G = 1 - \frac{G(\pi_1, \pi_2)}{G_{max}} \tag{1.13}$$

where $G_{max} = \max_{\pi_i \in \pi^N, \pi_j \in \pi^N} G(\pi_i, \pi_j)$ and π^N the set of all normalized possibility distributions on Ω.

1.3.2.2 Sangûesa et al's distance

In [21], authors proposed Sangûesa et al's distance, denoted by *distance*, and defined it as the U-uncertainty of distributions difference.

$$distance(\pi_1, \pi_2) = U(|\pi_1(\omega_i) - \pi_2(\omega_i)|) \quad \forall(\omega_i \in \Omega) \tag{1.14}$$

The Sangûesa et al's distance can be transformed to assess the similarity between two possibility distributions. This derived similarity measure, denoted by $S_{distance}$, can be defined as:

$$S_{distance} = 1 - \frac{distance(\pi_1, \pi_2)}{distance_{max}} \tag{1.15}$$

where $distance_{max} = \max_{\pi_i \in \pi^N, \pi_j \in \pi^N} distance(\pi_i, \pi_j)$ and π^N the set of all normalized possibility distributions on Ω.

1.3.2.3 Information divergence

The information divergence, denoted by D, is a way of comparing possibility distributions and it forms the analog of information divergence in the probabilistic case.

$$D(\pi_1|\pi_2) = \sum_{i=1}^{n} \pi_d(\omega_{\sigma(i)})[\Pi_1(A_{\sigma(i)} - \Pi_1(A_{\sigma(i+1)}] \tag{1.16}$$

where σ refers to permutation of indexes s.t.:

$\pi_d(\omega_{\sigma(i)}) \leq ... \leq \pi_d(\omega_{\sigma(n)})$ and $A_{\sigma(i)} = \{\omega_{\sigma(i)}, ..., \omega_{\sigma(n)}\}$ with i = 1..n and $A_{\sigma(n+1)} = \emptyset$.

The information divergence measure can be transformed to assess similarity between two possibility distributions. This derived similarity measure, denoted by $S_{distance}$, can be defined as:

$$S_D = 1 - D(\pi_1|\pi_2) \tag{1.17}$$

1.3.2.4 Minkowski distance and its derivatives

In the possibilistic case, we can use the well known Minkowski distance, denoted by D, which is a generalised metric that includes others as special cases of the generalised form :Manhattan , Euclidean and Maximum distance (or Chebyshev distance or chessboard distance) denoted respectively by D_M, D_E and D_C. Then we can define similarity measures derived from them, denoted respectively by S_M, S_E and S_C.

$$D = \sqrt[p]{\sum_{i=1}^{n} |\pi_1(\omega_i) - \pi_2(\omega_i)|^p} \tag{1.18}$$

Derivates of Minkowski distance are:

- If p = 1 , we define Manhattan distance.

$$D_M = \frac{\sum_{i=1}^{n} |\pi_1(\omega_i) - \pi_2(\omega_i)|}{n} \tag{1.19}$$

  The similarity measure S_M is defined as follows:

$$S_M = 1 - \frac{\sum_{i=1}^{n} |\pi_1(\omega_i) - \pi_2(\omega_i)|}{n} \tag{1.20}$$

- If p = 2 , we define the Euclidean distance.

$$D_E = \sqrt{\frac{\sum_{i=1}^{n} (\pi_1(\omega_i) - \pi_2(\omega_i))^2}{n}} \tag{1.21}$$

  The similarity measure S_E is defined as follows:

$$S_E = 1 - \sqrt{\frac{\sum_{i=1}^{n} (\pi_1(\omega_i) - \pi_2(\omega_i))^2}{n}} \tag{1.22}$$

- If we use the maximum function, we define the Maximum distance.

$$D_C = \max_{i=1}^{n} |\pi_1(\omega_i) - \pi_2(\omega_i)| \tag{1.23}$$

The similarity measure S_C is defined as follows:

$$S_C = 1 - \max_{i=1}^{n} |\pi_1(\omega_i) - \pi_2(\omega_i)| \tag{1.24}$$

1.3.2.5 Information affinity

In [15], authors have shown that measuring similarity between two possibility distributions depends on two main criteria distance and inconsistency and they define their measure information affinty, denoted by Aff.

The information affinity between two possibility distributions π_1 and π_2 is defined as follows:

$$Aff(\pi_1, \pi_2) = 1 - \frac{\kappa * d(\pi_1, \pi_2) + \lambda * Inc(\pi_1, \pi_2)}{\kappa + \lambda} \tag{1.25}$$

where $\kappa > 0$, $\lambda > 0$, $d(\pi_1, \pi_2)$ is either Euclidean distance or Manhattan distance between π_1 and π_2 given by equations (1.19), (1.21) and $Inc(\pi_1, \pi_2)$ is the degree of inconsistency between π_1 and π_2 given by equation (1.6).

In [14], the author has shown that his measure of similarity information affinity which is an extension of Euclidean and Manhattan and combines distance and inconsistency satisfies all properties listed above (1.1) - (1.10).

Example 1.5. *Let us consider the following possibility distributions over the same universe of discourse* $\Omega = \{\omega_1, \omega_2, \omega_3, \omega_4\}$*:* $\pi_1 = \{0.2, 0.9, 0.3, 1\}$ *and* $\pi_2 = \{0.1, 0.5, 1, 0\}$.

Let us compute the distance between the two distributions using distances listed above given by equations: (1.12), (1.14), (1.16), (1.19), (1.21) and (1.23), respectively:

- $G(\pi_1, \pi_2) = 2 * U\{1, 1, 0.9, 0.2\} - U\{0.2, 0.9, 0.3, 1\} - U\{0.1, 0.5, 1, 0\} = 2 * 1.6094 - 1.1584 - 0.5584 = 1.502$

- $distance(\pi_1, \pi_2) = U\{0.7, 0.4, 0.1\} = 0.9754$
- $D(\pi_1, \pi_2) = 0.1 * (1-1) + 0.4 * (1-1) + 0.7 * (1-1) + 1 * (1-0) = 1$
- $D_M(\pi_1, \pi_2) = \frac{0.1+0.4+0.7+1}{4} = 0.55$
- $D_E(\pi_1, \pi_2) = \sqrt{\frac{0.1^2+0.4^2+0.7^2+1^2}{4}} = 0.64$
- $D_C(\pi_1, \pi_2) = max(0.1, 0.4, 0.7, 1) = 1$

Let us compute the similarity degree between the two distributions using similarity measures listed above given by equations: (1.13), (1.15), (1.17), (1.20), (1.22), (1.24) and (1.25), respectively:

- $S_G(\pi_1, \pi_2) = 1 - \frac{G(\pi_1,\pi_2)}{2*log_2(4)-log_2(3)} = 0.37$
- $S_{distance}(\pi_1, \pi_2) = 1 - \frac{distance(\pi_1,\pi_2)}{log_2(4)} = 0.51$
- $S_D(\pi_1, \pi_2) = 1 - S_D(\pi_1, \pi_2) = 0$
- $S_M(\pi_1, \pi_2) = 1 - D_M(\pi_1, \pi_2) = 0.45$
- $S_E(\pi_1, \pi_2) = 1 - D_E(\pi_1, \pi_2) = 0.35$
- $S_C(\pi_1, \pi_2) = 1 - D_C(\pi_1, \pi_2) = 0$
- $Aff(\pi_1, \pi_2) = 1 - \frac{1*0.45+1*0.5}{1+} = 0.47$

Note that in the report, in examples in which we use information affinity, we use Manhattan distance, $\kappa = 1$ and $\lambda = 1$.

1.4 Conclusion

In this chapter we have presented basics of possibility theory, a non classical theory of uncertainty that offers an appropriate framework to handle uncertainty qualitatively and quantitatively. Then, we have introduced the notion

of distance and similarity and we have presented several measures that can be applied in the possibilistic case and their properties.

One similarity measure satisfy both basic and extended properties which is information affinity.

In Chapter 3, the notion of similarity measure will represent a key element in induction possibility distributions from dataset of sample cases and we will use information affinity measure.

Chapter 2

Learning possibilistic networks

2.1 Introduction

Possibilistic graphical models are marriage between possibility theory and graph theory. They are composed of a graph (undirected graph, tree, directed acyclic graph...) and possibility distributions. The success of possibilistic graphical models is due to its capacity to treat imprecision, uncertainty and complexity.

One of the well known possibilistic graphical models are possibilistic networks. Several researches concerning them have focused in drawing inference in order to derive evidence [1, 2, 3] and despite their popularity, few works were dedicated to learn them from datasets [3]. However, the learning task - if the data is available- is a promising area of research because constructing them by human experts is tedious and time consuming.

This chapter is organized as follows: Section 2.2 introduces briefly possibilistic networks. Section 2.3 presents learning possibilistic networks structure. we list several evaluation measures and methods existing in literature. Section 2.4 investigates the concept of induction a possibility distributions from a dataset of sample cases.

2.2 Possibilistic networks

Possibilistic networks are well known possibilistic graphical models and powerful tools used to deal with representation and processing of uncertain and imprecise information. They are denoted by ΠG and can be viewed as the counterpart of Bayesian networks [17, 20]. Possibilistic networks are composed of two main components : graphical and numerical ones.

In this section, we start by giving a general notation relative to possibilistic networks. Then, we will briefly present them in the qualitative and quantitative settings.

2.2.1 Notations

Let $V = \{V_1, V_2, ..., V_n\}$ be a set of variables describing such domain and E is a part of the Cartesian product $V \times V$. The graphical component of a possibilistic network is represented by a DAG (directed acyclic graph) and represented by the couple $G = (V, E)$ in which:

- Each node in V represents a variable.
- Each oriented edge or also called arc V_iV_j represents a dependency relation between two variables.
- V_1 is a parent of V_2 if there is an edge from V_1 to V_2. V_2 is called a child of V_1.
- pa_i denotes the set of parents of a node V_i.
- A path is defined as a sequence of nodes in V from one node to an other using arcs in E.
- A cycle is defined as a path which visits each node once and starts and ends with the same node. Possibilistic networks does not contain cycles.

The numerical component is the possibility distribution affected to each node in the context of its parents. It quantify uncertainty represented at each node by local conditional distribution : $\Pi(V_i|pa_i)$. The numerical component represents the quantification of the links between nodes. We mentioned in Chapter 1 two interpretations to the conditioning in the possibilistic case. So, naturally, there are two different ways to define possibilistic networks: qualitative and quantitative.

Example 2.1. *Figure(2.1) and table(2.1) illustrate an example of possibilistic networks.*

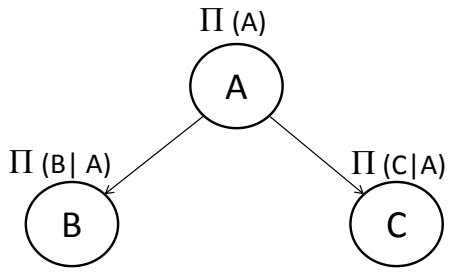

Figure 2.1: Graphical component of the possibilistic network

a	$\Pi(a)$	a	b	$\Pi(b\|a)$	a	c	$\Pi(c\|a)$
a_1	0.4	a_1	b_1	0.3	a_1	c_1	1
a_2	1	a_1	b_2	1	a_1	c_2	1
		a_2	b_1	0.4	a_2	c_1	0.4
		a_2	b_2	0.7	a_2	c_2	0.1

Table 2.1: Numerical component of the possibilistic network

2.2.2 Qualitative and quantitative possibilistic networks

Qualitative possibilistic networks, also called *min-based* possibilistic networks and denoted by ΠG_{min}, are based on the *min-based* conditioning expressed

by equation (1.9).

The joint distribution relative to *min-based* possibilistic networks, denoted by π_m, can be computed via the following *min-based* chain rule:

Definition 2.1. *(**min-based chain rule**) Given a min-based possibilistic network ΠG_m, the global joint possibility distribution over the variable set $V = \{V_1, V_2, ..., V_N\}$ can be expressed as the minimum of the N initial a priori and conditional possibilities via the following min-based chain rule:*

$$\pi_m(V_1, .., V_N) = \min_{i=1..N} \Pi(V_i \mid pa_i(V_i)). \tag{2.1}$$

Example 2.2. *Table(2.2) gives the joint possibility distribution relative to the possibilistic network given in example(2.1).*

a	b	c	$\pi_m(a \wedge b \wedge c)$	a	b	c	$\pi_m(a \wedge b \wedge c)$
a_1	b_1	c_1	0.3	a_2	b_1	c_1	0.4
a_1	b_1	c_2	0.3	a_2	b_1	c_2	0.1
a_1	b_2	c_1	0.4	a_2	b_2	c_1	0.4
a_1	b_2	c_2	0.4	a_2	b_2	c_2	0.1

Table 2.2: The joint possibility distribution

Quantitative possibilistic networks, also called *product-based* possibilistic networks and denoted by ΠG_*, are based on the *product-based* conditioning expressed by equation (1.10).

The joint distribution relative to *product-based* possibilistic networks, denoted by π_p, can be computed via the following *product-based* chain rule:

Definition 2.2. *(**Product-based chain rule**) Given a product-based possibilistic network ΠG_*, the global joint possibility distribution over the variable set $V = \{V_1, V_2, ..., V_N\}$ can be expressed as the product of the N initial a priori and conditional possibilities via the following product-based chain rule:*

$$\pi_p(V_1, ..., V_N) = \prod_{i=1..N} \Pi(V_i \mid pa_i(V_i)), \tag{2.2}$$

Example 2.3. *Table(2.3) gives the joint possibility distribution relative to the possibilistic network given in example(2.1).*

a	b	c	$\pi_p(a \wedge b \wedge c)$	a	b	c	$\pi_p(a \wedge b \wedge c)$
a_1	b_1	c_1	0.12	a_2	b_1	c_1	0.16
a_1	b_1	c_2	0.12	a_2	b_1	c_2	0.04
a_1	b_2	c_1	0.4	a_2	b_2	c_1	0.28
a_1	b_2	c_2	0.4	a_2	b_2	c_2	0.07

Table 2.3: The joint possibility distribution

2.3 Learning possibilistic networks

Despite the popularity of possibilistic networks and the number of researches focusing in propagation algorithms [1, 2, 3], few works have been proposed to learn them from data[3] and in most cases, they are constructed using expert opinions.

In this section, we give a brief introduction to approaches of learning graphical models structure originally proposed for Bayesian networks. Then, we focus in learning possibilistic networks.

2.3.1 Learning graphical models

Outside construction by experts, there are three basic approaches for learning graphical models structure from data :

- **Constraint based approach:** This approach consists in trying to cover all conditional independences existing in the graph then constructing it using these independences. Conditional independence tests reduce the number of candidate graphs. In fact, if one test fails, many graphs would be excluded. This approach presents two drawbacks.

First, the number of conditional independences in a graph is enormous. Second, it supposes that all conditional independences are necessarily found.

- **Score based approach:** The score based approach looks for structure that maximizes a certain score, or looks for the best structures and combines them. This approach corresponds to construction based on the strength of dependences between variables existing in the graph using the strong assumption that two adjacent variables are more dependent than two non adjacent ones. This is a heuristic, but often highly successful approach, because in most cases, it gives good results.

- **Hybrid method:** This approach combines advantages of the two already listed methods. It consists in two steps. First, a local search uses independences test in order to provide a neighborhood containing all interesting conditional dependences and independences. Second, a global optimization is performed in order to search in the space of candidate graphs by restricting to only conditional dependences already found.

2.3.2 Learning possibilistic networks

Unlike the probabilistic case where several works have been proposed to learn Bayesian networks, few works were interested to learn possibilistic networks. Learning parameters still poses several problems. That's why, in most cases, they are learned using experts opinions. In the following part, we talk about learning structure and we present methods proposed by Borgelt[3] which belong to score based approach and consist in a search method in the space of possible graphs given a dataset guided by a possibilistic evaluation measure as illustrated in figure(2.2).

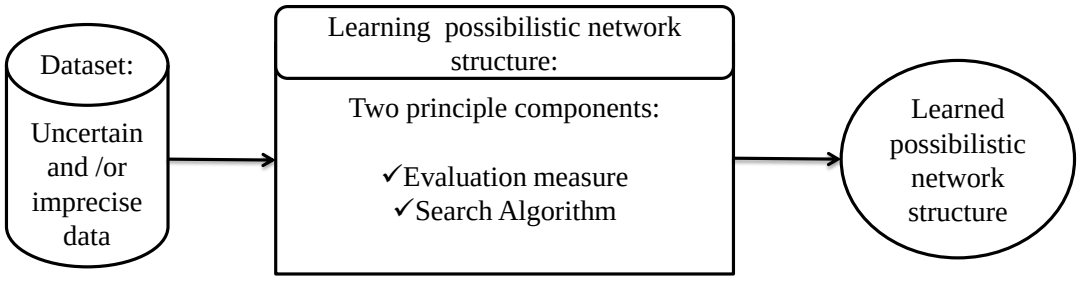

Figure 2.2: Learning possibilistic networks structure

2.3.3 Evaluation measures

Evaluation measures are used to compare candidate graphs, with assessing the quality of each one of them w.r.t given data set, in order to choose the best one which fits well the dataset. Evaluation measures can be classified in two categories: local measures and global measures.

Local measures are based on calculating the strength of conditional dependences between variables. In fact, we can compute local scores to subnetworks or simply to one edge (score between two variables). It is permitted due to the decomposability property of an evaluation measure permitting to express it as sum of scores at each local node. Unlike local measure, global measures evaluate the quality of the whole graph given a dataset and can not be decomposed.

Obviously, the objective of any measure either local or global is finding the graph which represents strongest dependences existing between two or more variables. Several measures have been presented in literature [9, 11] and showed their efficiency. We present principle ones that can be applied in the possibilistic case.

All evaluation measures listed in th following part are derived either from relational or, often by analogy, from probabilistic evaluation measures. In the following, let A and B two variables , $dom(A) = \{a_1, a_2, ..., a_n\}$ and

$dom(B) = \{b_1, b_2, ..., b_n\}$ their corresponding domains.

2.3.3.1 Specificity gain

The specificity gain, named S_{gain}, can be derived using the α-cut concept [18] or using U-uncertainty measure. The notion of α-cut view is used in fuzzy set theory in order to decompose a fuzzy set. In the possibilistic case, possibility distributions are decomposed forming an α-cut relation allowing the definition of the $[\Pi]_\alpha$ with $\alpha \in [0, 1]$. Thus, all tuples whose possibility degrees greater than or equal to α take 1 and other tuples take 0.

The underlying idea of specificity gain is computing Hartley information gain for each α-cut of the possibility distribution. These values are then aggregated by integrating over all values of α. For more details, see [5, 7]. The specificity gain is defined as follows:

$$\begin{aligned} S_{gain}(A, B) = \int_0^{sup\Pi_{AB}} & log_2(\sum_{a \in dom(A)} [\Pi_A]_\alpha(a)) \\ & + log_2(\sum_{b \in dom(B)} [\Pi_B]_\alpha(b)) \\ & - log_2 \sum_{\substack{a \in dom(A) \\ b \in dom(B)}} [\Pi_{AB}]_\alpha(a, b))\ d\alpha \end{aligned} \tag{2.3}$$

This formula can also be obtained using the notion of U-uncertainty or also called non-specificity, denoted by nsp, as known in the possibility theory [4]. Let Ω be the universe of discourse of Π and $\omega \in \Omega$. The nsp is defined as:

$$nsp(\Pi) = \int_0^{sup\Pi} log_2(\sum_{\omega \in \Omega} [\Pi]_\alpha(\omega)) d\alpha \tag{2.4}$$

And the specificity gain is given by:

$$S_{gain}(A, B) = nsp(\Pi_A) + nsp(\Pi_B) - nsp(\Pi_{AB}) \tag{2.5}$$

Certain normalizations of the specificity gain have been proposed: the specificity gain ratio named S_{gr} and two symmetric specificity gain ratios denoted

$S_{gr}^{(1)}$ and $S_{gr}^{(2)}$:

$$S_{gr}(A,B) = \frac{S_{gain}(A,B)}{nsp(\Pi_B)} = \frac{nsp(\Pi_A) + nsp(\Pi_B) - nsp(\Pi_{AB})}{nsp(\Pi_B)} \tag{2.6}$$

$$S_{gr}^{(1)} = \frac{S_{gain}(A,B)}{nsp(\Pi_{AB})} \tag{2.7}$$

$$S_{gr}^{(2)} = \frac{S_{gain}(A,B)}{nsp(\Pi_A) + nsp(\Pi_B)} \tag{2.8}$$

Another variant of the specificity gain which is known by giving a direct indication of the strength of dependence of two variables due to the introducing of conditional relations. This measure is named S_{cgain} and defined as:

$$S_{cgain} = \sum_{b \in dom(B)} \int_0^{\Pi_B(b)} \frac{[\Pi_B]_\alpha(b)}{\sum\limits_{b \in dom(B)} [\Pi_B]_\alpha(b)} log_2 \frac{\sum\limits_{a \in dom(A)} [\Pi_A]_\alpha(a)}{\sum\limits_{b \in dom(B)} [\Pi_{A|B}]_\alpha(a|b) d\alpha} \tag{2.9}$$

2.3.3.2 Possibilistic mutual information

The possibilistic mutual information named $d_{mi}(A,B)$ is based on the cross entropy , denoted by $I_{mutual}^{(shanon)}(A,B)$, introduced by Shanon and expressed by:

$$I_{mutual}^{(shanon)}(A,B) = \sum_{\substack{a \in dom(A) \\ b \in dom(B)}} p_{AB}(a,b).log_2 \frac{p_{AB}(a,b)}{p_A(a).p_B(b)} \tag{2.10}$$

By forming the analog of the mutual information [5], we obtain:

$$d_{mi}(A,B) = - \sum_{\substack{a \in dom(A) \\ b \in dom(B)}} \Pi_{AB}(a,b).log_2 \frac{\Pi_{AB}(a,b)}{\min\left(\Pi_A(a), \Pi_B(b)\right)} \tag{2.11}$$

Another variant has been proposed in [11] denoted by Γ_L:

$$\Gamma_L(A,B) = - \sum_{\substack{a \in dom(A) \\ b \in dom(B)}} \Pi_{AB}(a,b).log[1 + \Pi_{AB}(a,b) - min(\Pi_A(a)), \Pi_B(b)] \tag{2.12}$$

2.3.3.3 Possibilistic χ^2 measure

This measure is also based on the underlying idea of the mutual information described above and it is the analog of the probabilistic χ^2 measure. The possibilistic χ^2 measure is named d_{χ^2} and defined as in [5]:

$$d_{\chi^2} = \sum_{\substack{a \in dom(A) \\ b \in dom(B)}} \frac{(min(\Pi_A(a), \Pi_B(b) - \Pi_{AB}(a,b))^2}{min(\Pi_A(a), \Pi_B(b))} \tag{2.13}$$

Another measure which can be cited in this context is the weighted sum of squared differences named $d_{diff}(A,B)$ and defined as follows:

$$d_{diff}(A,B) = \sum_{\substack{a \in dom(A) \\ b \in dom(B)}} (\min(\Pi_A(a), \Pi_B(b)) - \Pi_{AB}(a,b))^2 \tag{2.14}$$

2.3.3.4 Weighted sum of possibility degrees

The weighted sum of possibility degrees named Q(G) is a global evaluation measure. Unlike the three evaluation measures described above, it does not support decomposition. A global measure compares the distribution described by a graph to the one that is induced by the dataset to learn from. Obviously, the number of possible tuples (all combinations of variables) in such domain can be enormous. So, this measure proposes to form a proper subset containing only tuples t in dataset D and compare the graph to this subset. The underlying idea of weighted sum of possibility degrees is summing possibility degrees of possible tuples weighted with their frequency (w(t)) in the dataset [6].

$$Q(G) = \underset{t \in D}{w}(t).\Pi_G(t) \tag{2.15}$$

2.3.4 Learning possibilistic networks structure

The objective of a learning method is giving as an output the best graphical model that fits well the dataset. In the case of possibilistic networks, it

consists in searching in the space of the domain and finding the best DAG. However, searching and evaluating all candidate graphs tends to be infeasible in high-dimensional domain in which the number of graphs is huge. In the following part, we will introduce several search methods used in learning graphical models and tested in the case of possibilistic networks.

2.3.4.1 MWST Kruskal algorithm

The kruskal algorithm gives as an output a maximum weight spanning tree (MWST). The search of the structure is based on weights assigned to edges and calculated with an evaluation measure (local scores). The obtained structure is a subgraph in the form of an undirected tree. To transform it to a directed graph, one possible solution is using Depth-first search which is an algorithm for traversing or searching a tree and adding orientation to edges.

The kruskal algorithm starts from a set of trees passing by nodes, denoted by T, and merges them. It constructs rapidly a tree but it connects all nodes(variables) even if the weight between nodes is too small. Thus, it can add unnecessary nodes.

2.3.4.2 Greedy parent search algorithm

The greedy parent search algorithm requires a topological order to reduce the search space: If the node A is listed before the node B, there isn't an edge from B to A. It consists in finding the best set of parents for a node in order to maximize the score of subnetworks and combine them. It is based on measuring the strength of dependence between variables and candidate parent variables.

Evaluation measures compute the local score of the parentless child. The following step consists in choosing the best parent from all candidate parents. If its score is greater than the score of parentless child, the parent which has the highest score is chosen and added to this child. Then, the child and the parent are combined into a pseudo-variable and the score is recomputed.

Algorithm 2.1: MWST

Data: In: set of ordered edges(descending) A
Result: B the resulting MWST
begin
  foreach V_i **do**
    $T(V_i) \leftarrow \{V_i\}$
  $B \leftarrow \emptyset$
  foreach $(V_i, V_j) \in A$ **do**
    if $T(V_i) \neq T(V_j)$ **then**
      $B \leftarrow B \cup (V_i, V_j)$
      $T' \leftarrow T(V_i) \cup T(V_j)$
      $T(V_i) \leftarrow T'$
      $T(V_j) \leftarrow T'$
  return B
end

This procedure is stopped when one of termination criteria is met (no more parent, the maximum number of parents, denoted by u, is reached). Two drawbacks can be listed to this algorithm. First,the choice of the order may be a complex task. Second, it can converge to a local optimum with selecting a wrong set of parent variables.

2.3.4.3 Simulated annealing

The simulated annealing algorithm is a guided random search method because it evaluates a random solution using a global evaluation measure ($E(s)$). The idea underlying the simulated annealing algorithm is ameliorating a randomly generated candidate solution (S_0). It is an iterative procedure which is composed of two main components: generating a solution (S) from candidate ones and evaluating it.

The objective of the evaluating step is testing whether the generated solution (S_{new}) is better than the last generated one. If it is better, it is chosen and we replace the last solution with the new one. If it is worse, the solu-

Algorithm 2.2: Greedy parent search

Data: In: local score g, u In: Data
begin
  for $i \in \{1, \ldots, n\}$ **do**
  $pa_i \leftarrow \emptyset$
  $g_{old} \leftarrow g(i, pa_i)$
  $Ok \leftarrow true$
  repeat
    $find\ V_j \in Pred(V_i) \setminus pa_i$ **maximizing** $g(i, pa_i \cup V_j)$
    if $g_{new} > g_{old}$ **then**
      $g_{old} \leftarrow g_{new}$
      $pa_i \leftarrow pa_i \cup V_j$
    else
      $ok \leftarrow False$
  until $Ok\ and\ \|pa_i\| < u$;
end

tion is accepted with certain probability. The simulated annealing algorithm takes as input: emax(expected maximum value of evaluation measure) , T: temperature, Kmax: maximum number of iterations. Note that in our case only the weighted sum of possibility degrees is a global measure.

2.4 Database-induced possibility distributions

Learning possibilistic networks from data requires a method to extract possibility distributions from datasets of sample cases. In fact, it forms a very important step in the learning task because without calculating marginal distributions, it is impossible to learn possibilistic networks structure. Few works was dedicated to this area of research. In [8], authors proposed a method based on calculating the maximum projection from imprecise dataset (dataset that contains set valued information).

We first introduce some basic notions that will be used in order to detail this method.

Algorithm 2.3: Simulated annealing

```
Data: In: T, Kmax, emax, Data , E
Result: Sbest
begin
    S ← S_0
    e ← E(s)
    Sbest ← S
    Ebest ← e
    K ← 0
    while (K < Kmax and e > emax) do
        Snew ← neighboor(s)
        Enew ← E(Snew)
        if e_new < e or p(e, enew, T) > random() then
            S ← Snew
            e ← enew
            K ← K + 1
    return Sbest
end
```

Definition 2.3. *Let $U = \{A_1, A_2, ..., A_n\}$ be the set of attributes and $dom(A_i)$ their corresponding domains. The precise tuple t_U is a mapping over U that assigns to every attribute A_i one value:*

$$t_U^{precise} : U \rightarrow \bigcup_{A \in U} dom(A)$$

Example 2.4. *Let A, B and C be three attributes and $dom(A) = \{a_1, a_2\}$, $dom(B) = \{b_1, b_2\}$ and $dom(C) = \{c_1, c_2\}$ their corresponding domains. A possible precise tuple is $t = (A \longmapsto \{a_1\},\ B \longmapsto \{b_2\}, C \longmapsto \{c_2\})$.*

Definition 2.4. *Let $U = \{A_1, A_2, ..., A_n\}$ be the set of attributes and $dom(A_i)$ their corresponding domains. The imprecise tuple t_U is a mapping over U that may assigns to at least one attribute A_i more than one value:*

$$t_U^{imprecise} : U \rightarrow \bigcup_{A \in U} 2^{dom(A)}$$

Example 2.5. *Let A, B and C be three attributes and their corresponding domains $dom(A) = \{a_1, a_2, a_3, a_4\}$, $dom(B) = \{b_1, b_2, b_3\}$ and $dom(C) =$*

$\{c_1, c_2\}$. *A possible imprecise tuple is* $t = (A \longmapsto \{a_1, a_3\},\ B \longmapsto \{b_1, b_2\}, C \longmapsto \{c_2\})$.

Definition 2.5. *A database* D_U *over* U *is a pair* (R_U, w_{R_U}) *where* R_U *is the relation over* U *containing all possible tuples in* D *and* w_{R_U} *is a mapping from the tuple (precise and imprecise) in* R_U *to* $\mathbb{N}$ *which indicates the number of occurrence of the tuple* t.

Before extracting marginal distributions, a preprocessing phase *'closure under tuple intersection'* must be performed. It consists in adding every tuple that corresponds to an intersection of others existing in the original dataset. The weight of this resulting tuple is the sum of weights of the two tuples that intersect. The new added tuple is as least as specific as (contained in) the two others that intersect.

Example 2.6. *let* $U = \{A, B\}$ *be a set of attributes over* U *with their corresponding domains* $dom(A) = \{a_1, a_2, a_3\}$ *and* $dom(B) = \{b_1, b_2, b_3\}$.
Let us consider an example of dataset over U*:*
$t_1 = (\{a_1, a_2, a_3\}, \{b_3\}) : 1$
$t_2 = (\{a_1, a_2\}, \{b_2, b_3\}) : 1$
$t_3 = (\{a_3, a_4\}, \{b_1\})$ *:1*
The two first tuples intersect in $t_4 = (\{a_1, a_2\}, \{b_3\})$. *So, we must add* t_4 *with a weight = 2 (sum of weights of* t_1*) and* t_2. t_4 *is said* at least as specific as t_1 *and* t_2.

Now to estimate marginal possibility distributions, we should compute maximum projection. We take for each precise tuple t_X of the subspace, defined by a set X of attributes, the maximum of weights of tuples t_U in the database whose projection to X is equal to t_X. Then, we divide by the sum of all weights in the original dataset.

$$\forall t_X : \qquad \Pi_X(t_X) = \max_{A \in U - X} \Pi_U^{(D)}(t_U) = \frac{\max_{A \in U - X} w_R(t_U)}{\sum_{s \in R} w_R(s)} \tag{2.16}$$

Note that this method was be applied in real world dataset with missing values.

Example 2.7. *Let us consider the dataset illustrated by table(2.4). Note that * indicates a missing value(i.e. all values of the attribute are possible):*

tuples			*frequency*
a_1	b_2	c_3	3
a_3	b_1	c_2	4
a_1	b_2	c_1	1
a_2	b_1	c_2	2
a_3	b_1	c_3	1
$*$	$*$	c_3	1
a_2	$*$	c_2	1

Table 2.4: An example of imprecise dataset

After preprocessing this dataset, we obtain the dataset illustrated by table(2.5): Finally, we compute possibility distributions:

tuples			*frequency*
a_1	b_2	c_3	4
a_3	b_1	c_2	4
a_1	b_2	c_1	1
a_2	b_1	c_2	3
a_3	b_1	c_3	2
$*$	$*$	c_3	1
a_2	$*$	c_2	1

Table 2.5: The preprocessed dataset

$\Pi(a1) = 0.3,$
$\Pi(a2) = 0.23,$
$\Pi(a3) = 0.3,$
$\Pi(b1) = 0.3,$
$\Pi(b2) = 0.3,$
$\Pi(c1) = 0.07,$

$\Pi(c2) = 0.3$,
$\Pi(c3) = 0.3$.

Now we can apply for example the greedy parent search algorithm(2.2) with the possibilistic mutual information, expressed by equation(2.11), as local evaluation measure for the database illustrated by table(2.4). The topological order used by the algorithm is: A,B,C. We then obtain the possibilistic network structure shown in figure(2.3).

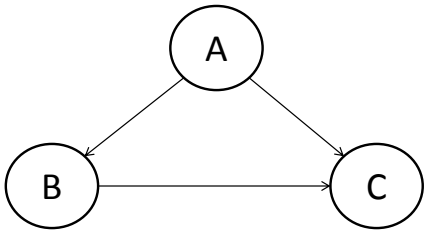

Figure 2.3: The resulted possibilistic network structure

The method described above for computing maximum projection is applicable only to datasets that contain set valued information because it is based on the context model interpretation [12] and was be tested with a real database containing missing value which can be not considered as true set valued information. Moreover, it treats only the imprecision concept designed by possibility theory. None of the tested dataset contains uncertain and imprecise information which is the key element of information representation in possibility theory.

Moreover, the preprocessing phase performed in the method presents some drawbacks. It valorizes imprecise information more than precise ones. In fact, if the tuple is imprecise, it will be affected to all tuples containing in it while the precise tuple is taken in account once. Furthermore, if the dataset contain only imprecise tuples (imprecision caused by missing values), computing closure under tuple intersection will be very expensive because because almost all possible precise and imprecise tuples will be added.

2.5 Conclusion

In this chapter, we have focused in learning possibilistic networks from dataset of sample cases. First, we have introduced them in quantitative and qualitative settings. Then, we have presented several methods that were proposed in literature to learn possibilistic networks structure. We listed many evaluation measures and search method algorithms that can be applied. In the last part of the chapter, we have explain a method used to extract marginal distributions from imprecise dataset of sample cases.

Chapter 3

New learning possibilistic networks structure algorithm

3.1 Introduction

In real life problems, we are drowning in data with high and low quality which is definitely related to the ways that the datasets are used. In our vision, we should always favor certain and precise data contrary to Borgelt's method in which more the information is imprecise more it is well represented. Our aim is knowledge extraction based mainly on data with high quality but without neglecting imperfect data because high quality information provide high quality knowledge.

Moreover, possibility theory has the ability to design two frequent sorts of imperfection: uncertainty and imprecision but Borgelt was interested only to imprecision. We have thought to propose a method that deals with these two concepts in order to benefit from the simplicity of the way we use to express opinions in possibilistic framework. Our objective is learning possibilistic networks from possibilistic datasets but the first difficulty we face is how we can induce possibility distributions from uncertain and imprecise data taking into account the importance of information quality, uncertainty and

imprecision.

This chapter is organized as follows: In section 3.2, we present several definitions we use in the chapter. Section 3.3 is devoted to representation of dataset preprocessing phase. In section 3.4, we present maximum projection method used to extract marginal possibility distributions. Then, in section 3.5, we compare our approach with existing method. Section 3.6 details learning possibilistic networks structure from possibilistic dataset and section 3.7 presents a method to evaluate it. Finally, in section 3.8, we present the implementation scheme. Section 3.9 is devoted to possible form of possibilistic datasets that can not be well treated with our approach to which we propose a possible solution.

3.2 Definitions

Before detailing our approach, we start by defining some notions that will be used in the remaining.

Definition 3.1. *Let $U = \{A_1, A_2, ..., A_n\}$ be the set of attributes and a_i the domains of attributes. The certain tuple $t_U^{certain}$ is the union over U of the mapping that assigns to one value of every attribute A_i 1 and 0 to other values i.e. A_i satisfy the case of complete knowledge:*

$$t_U^{certain} : U \rightarrow \bigcup_{\substack{A_i \in U \\ a_i \in A_i}} \pi(a_i)$$

Example 3.1. *Let A, B and C be three attributes and $dom(A) = \{a_1, a_2, a_3\}$, $dom(B) = \{b_1, b_2\}$ and $dom(C) = \{c_1, c_2, c_3\}$ their corresponding domains. A possible certain tuple is:*
$t = \{\pi(a_1), \pi(a_2), \pi(a_3), \pi(b_1), \pi(b_2), \pi(c_1), \pi(c_2), \pi(c_3)\}$
$= \{0, 0, 1, 0, 1, 0, 1, 0\}$.

Definition 3.2. *Let $U = \{A_1, A_2, ..., A_n\}$ be the set of attributes and a_i the domains of attributes. The uncertain tuple $t_U^{uncertain}$ is the union over*

U of the mapping that assigns to each value of every attribute A_i a number $\in$ [0, 1]:

$$t_U^{uncertain} : U \rightarrow \bigcup_{\substack{A_i \in U \\ a_i \in A_i}} \pi(a_i)$$

Example 3.2. *Let A, B and C be three attributes and $dom(A) = \{a_1, a_2, a_3\}$, $dom(B) = \{b_1, b_2\}$ and $dom(C) = \{c_1, c_2, c_3\}$ their corresponding domains. A possible uncertain tuple is:*
$t = \{\pi(a_1), \pi(a_2), \pi(a_3), \pi(b_1), \pi(b_2), \pi(c_1), \pi(c_2), \pi(c_3)\}$
$= \{0.2, 0.8, 1, 0.5, 1, 0, 1, 0.9\}$.

Definition 3.3. *A possibilistic training set is a collection of certain and uncertain tuples. A certain (resp. uncertain) training set is a part of the training set which contains only certain (resp. uncertain) tuples.*

Example 3.3. *Table(3.1) presents an example of a possibilistic training set. For instance, the five last tuples are uncertain and the others are certain.*

3.3 Dataset preprocessing phase

The first phase of our approach is preprocessing the training set. It consists as illustrated in figure(3.1) of two steps:

1. Extraction of distinct certain tuples from training set and computing their frequencies. Note that, we remove all fully uncertain tuples (containing in all attributes total ignorance e.g tuple = {11111111})

2. Computing the maximum similarity between each uncertain tuple and certain ones using the information affinity as measure of similarity given by equation(1.25) in order to affect it to the nearest certain tuple. Note that we discard similarities lower than certain threshold and we have chosen the information affinity because this measure satisfies both basic and extended properties.

A			B		C		
a_1	a_2	a_3	b_1	b_2	c_1	c_2	c_3
1	0	0	0	1	0	0	1
1	0	0	0	1	0	0	1
1	0	0	0	1	0	0	1
0	0	1	1	0	0	1	0
0	0	1	1	0	0	1	0
0	0	1	1	0	0	1	0
0	0	1	1	0	0	1	0
1	0	0	0	1	1	0	0
0	1	0	1	0	0	1	0
0	1	0	1	0	0	1	0
0	0	1	1	0	0	0	1
0.1	0	1	0.7	1	0.5	1	0.3
0	1	0	1	0.8	1	0.6	0.2
1	0	0.5	1	0.2	1	0	0
1	1	1	1	1	0	0	1
0	1	0	1	1	0	1	0

Table 3.1: An example of uncertain training set

Step1: Extraction of certain tuples: Given a training set, the certain and uncertain training sets are obtained using algorithm(3.1):

Example 3.4. *Let us reconsider the training set of example(3.3). After applying this step of our approach, we obtain values illustrated in tables(3.2) and (3.3):*

Step2: Computing similarity: Given certain and uncertain training sets, we compute the maximum of similarity between each uncertain and all certain tuples as shown by algorithm(3.2):

Example 3.5. *Let us consider the uncertain and certain training set of example(3.4), we compute the similarity between tuples using information*

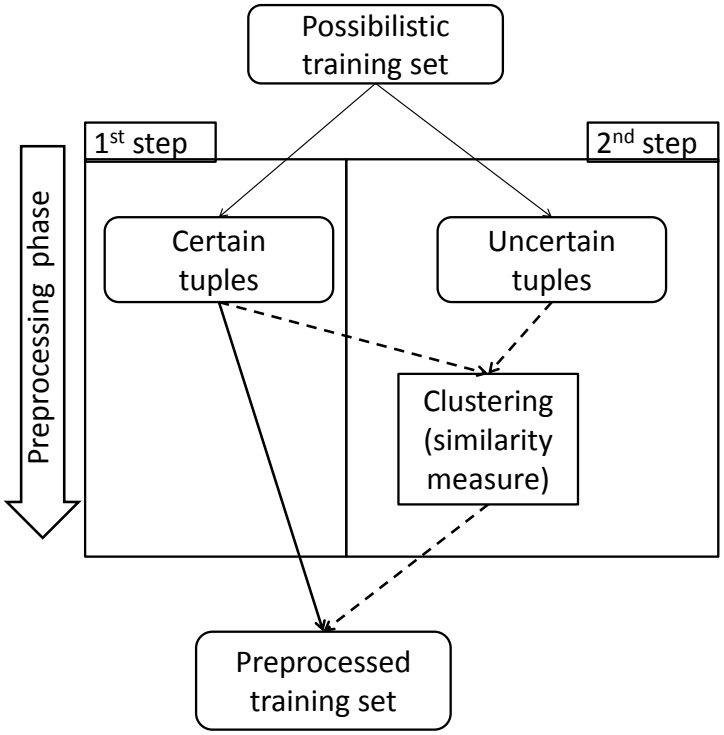

Figure 3.1: Preprocessing phase

affinity. The obtained values are illustrated by table(3.4). Uncertain tuples are presented by their numbers(1..5) as in table(3.3).

When we compute the similarity between tuples to affect uncertain tuples to the nearest certain one, we can be in one of these situations:

- The number of the nearest tuple = 1: This is illustrated in example(3.5) by tuples 1, 2 , 3 and 5. We add to the frequency of the nearest certain tuple the value of the similarity between it and the considered uncertain tuple.
- The number of the nearest tuple > 1: This is illustrated in example (3.5) by tuple 4 (case of equal similarities). To handle this case, we propose two solutions:

Algorithm 3.1: Extraction of certain and uncertain training sets

```
Data: In: training set, U set of attributes
Result: certain, uncertain
begin
    certain = ∅
    uncertain = ∅
    for j = 1..n do
        if t_j is fully uncertain then
            Remove(t_j)
        else
            if t_j is certain then
                add(t_j, certain)
            else
                add(t_j, uncertain)
        return certain, uncertain
end
```

1. Dispatching method.

2. Max frequency method.

3.3.1 Dispatching method

If for a given uncertain tuple, we obtain n nearest certain ones. The idea is to dispatch the maximum similarity between all nearest tuples as shown by algorithm(3.3):

Example 3.6. *Let us reconsider examples(3.4) and (3.5) and we use results illustrated by tables(3.2),(3.3) and 3.4). After applying Dispatching method, we obtain frequencies illustrated in table(3.5).*

Certain tuples	*frequency*
$a_1b_2c_3$	3
$a_3b_1c_2$	4
$a_1b_2c_1$	1
$a_2b_1c_2$	2
$a_3b_1c_3$	1

Table 3.2: Certain tuples and their frequencies

	Uncertain tuples							
	A			*B*		*C*		
N	a_1	a_2	a_3	b_1	b_2	c_1	c_2	c_3
1	0.1	0	1	0.7	1	0.5	1	0.3
2	0	1	0	1	0.8	1	0.6	0.2
3	1	0	0.5	1	0.2	1	0	0
4	1	1	1	1	1	0	0	1
5	0	1	0	1	1	0	1	0

Table 3.3: Uncertain tuples

3.3.2 Maximum frequency method

If for a given uncertain tuple, we obtain n nearest certain ones. The idea is to search in nearest tuples the one which has the highest frequency in the dataset and we affect the uncertain tuple as shown by algorithm(3.4). The intuition behind this method is , more the tuple occurs more it is possible.

Example 3.7. *Let us reconsider examples(3.4) and (3.5) and we use results illustrated by tables(3.2),(3.3) and 3.4). After applying Maximum frequency method, we obtain frequencies illustrated in table(3.5).*

Algorithm 3.2: Computing similarity

Data: In: certain, uncertain
Result: $max - sim$
begin
  foreach $t_u \in uncertain$ **do**
    foreach $t_c \in certain$ **do**
      $max - sim = maximum(Aff(t_u, t_c))$
  return $max - sim$
end

Uncertain tuples	*Certain tuples*				
	$a_1b_2c_3$	$a_3b_1c_2$	$a_1b_2c_1$	$a_2b_1c_2$	$a_3b_1c_3$
1	0.7	0.86	0.72	0.73	0.77
2	0.55	0.72	0.75	0.85	0.67
3	0.79	0.76	0.85	0.7	0.76
4	0.81	0.68	0.68	0.68	0.81
5	0.75	0.81	0.68	0.93	0.68

Table 3.4: Similarities between certain and uncertain tuples

3.4 Maximum projection

To extract distributions from the preprocessed training set (containing only certain tuples and their frequencies), we should determine the max of elementary degrees of possibility over all values of all other attributes after a projection to the considered subspace. This is done by taking the max for each precise tuple t_X of the subspace defined by the set X of attributes the frequencies of all tuples t_U in the training set in which t_X forms a part of them.

$$\forall t_X : \qquad \Pi_X(t_X) = \max_{A \in U-X} \Pi_U(t_U) = \frac{\max_{A \in U-X} frequency(t_U)}{\sum_{s \in D} frequency(s)} \tag{3.1}$$

After inducing initial possibility distributions, we can normalize them by dividing obtained ones for each attribute by the maximum of possibility

Algorithm 3.3: Dispatching method

Data: In: certain, uncertain, max-sim
Result: *certain*
begin
  foreach $t_u \in uncertain$ **do**
    $add(t_u, NearestTuple(t_u, certain), \frac{max-sim}{|NearestTuples|})$
  return *certain*
end

Certain tuples	$frequency + similarity$
$a_1b_2c_3$	$3 + 0.405$
$a_3b_1c_2$	$4 + 0.86$
$a_1b_2c_1$	$1 + 0.85$
$a_2b_1c_2$	$2 + 0.85 + 0.93$
$a_3b_1c_3$	$1 + 0.405$

Table 3.5: Affecting uncertain tuples using Dispatching method

distributions Π_{max}. They are denoted by (Π_n) and defined as follows:

$$\Pi_n = \frac{\Pi_X(t_X)}{\Pi_{max}} \tag{3.2}$$

To summarize, we obtain normalized possibility distributions using algorithm(3.5):

Example 3.8. *Let us reconsider results of example(3.6) illustrated by table(3.5). we obtain values illustrated by table(3.7):*

Now, we can extract possibility distributions:
Initial distributions:
$\Pi(a_1) = 0.22$,
$\Pi(a_2) = 0.24$,
$\Pi(a_3) = 0.31$,
$\Pi(b_1) = 0.31$,
$\Pi(b_2) = 0.22$,

Algorithm 3.4: Maximum frequency method

Data: In: certain, uncertain, max-sim
Result: *certain*
begin
  foreach $t_u \in uncertain$ **do**
    if $|nearest - tuple| = 1$ **then**
      $add(t_u, nearest - tuple(t_u, certain), max - sim)$
    else
      $add(t_u, maxFreq[NearestTuple(t_u, certain), max - sim])$
  return *certain*
end

Certain tuples	$frequency + similarity$
$a_1b_2c_3$	$3 + 0.81$
$a_3b_1c_2$	$4 + 0.86$
$a_1b_2c_1$	$1 + 0.85$
$a_2b_1c_2$	$2 + 0.85 + 0.93$
$a_3b_1c_3$	1

Table 3.6: Affecting uncertain tuples using Maximum frequency method

$\Pi(c_1) = 0.12$,
$\Pi(c_2) = 0.31$,
$\Pi(c_3) = 0.22$.
Normalized distributions:
$\Pi_n(a_1) = 0.7$,
$\Pi_n(a_2) = 0.77$,
$\Pi_n(a_3) = 1$,
$\Pi_n(b_1) = 1$,
$\Pi_n(b_2) = 0.7$,
$\Pi_n(c_1) = 0.38$,
$\Pi_n(c_2) = 1$,
$\Pi_n(c_3) = 0.7$.

Example 3.9. *Let us now reconsider results of example(3.7) illustrated by*

Algorithm 3.5: Computing normalized possibility distributions

Data: In: certain, U:set of attributes
Result: $marginal - distributions$
begin
  foreach $A \in U$ **do**
    foreach $a_i \in A$ **do**
      $marg - distr(a_i) = max - projection(certain, a_i)$
      $normalize(marg - distr)$
  return $marginal - distributions$
end

Certain tuples	*maximum projection*
$a_1b_2c_3$	0.22
$a_3b_1c_2$	0.31
$a_1b_2c_1$	0.12
$a_2b_1c_2$	0.24
$a_3b_1c_3$	0.09

Table 3.7: Computing maximum projection (case Dispatching method)

table(3.6). We obtain values illustrated by table(3.7):

Now, we can extract possibility distributions:
Initial distributions:
$\Pi(a_1) = 0.24$,
$\Pi(a_2) = 0.24$,
$\Pi(a_3) = 0.31$,
$\Pi(b_1) = 0.31$,
$\Pi(b_2) = 0.24$,
$\Pi(c_1) = 0.12$,
$\Pi(c_2) = 0.31$,
$\Pi(c_3) = 0.24$.
Normalized distributions:
$\Pi_n(a_1) = 0.77$,

Certain tuples	*max projection*
$a_1b_2c_3$	0.24
$a_3b_1c_2$	0.31
$a_1b_2c_1$	0.12
$a_2b_1c_2$	0.24
$a_3b_1c_3$	0.06

Table 3.8: Computing maximum projection (case maximum frequency method)

$\Pi_n(a_2) = 0.77$,
$\Pi_n(a_3) = 1$,
$\Pi_n(b_1) = 1$,
$\Pi_n(b_2) = 0.77$,
$\Pi_n(c_1) = 0.38$,
$\Pi_n(c_2) = 1$,
$\Pi_n(c_3) = 0.77$.

3.5 Borgelt's method vs our approach

Borgelt's method is detailed in the last chapter in section(2.4). This method deals with imprecise dataset which can be a particular case of datasets we manipulate. That's why in this section, we apply our approach to this particular case. In fact, we apply our approach to the dataset illustrated by table(2.4) which illustrates Borgelt's method. Then we will explain the difference between the two methods and present advantages of using our approach.

Example 3.10. *Let us reconsider the dataset in example(2.7). First, we preprocess the dataset. We obtain values illustrated by tables(3.9) and (3.10):*

Then we compute similarities between certain and uncertain tuples. We obtain values illustrated by table(3.11).

If we apply dispatching method, we obtain values illustrated by table(3.12):

Certain tuples	*frequency*
$a_1b_2c_3$	3
$a_3b_1c_2$	4
$a_1b_2c_1$	1
$a_2b_1c_2$	2
$a_3b_1c_3$	1

Table 3.9: Uncertain tuples and their frequencies

	Uncertain tuples							
	A			B		C		
N	a_1	a_2	a_3	b_1	b_2	c_1	c_2	c_3
1	1	1	1	1	1	0	0	1
2	0	1	0	1	1	0	1	0

Table 3.10: Uncertain tuples

Now, we compute marginal distributions:
Initial distributions :
$\Pi(a_1) = 0.26$,
$\Pi(a_2) = 0.22$,
$\Pi(a_3) = 0.31$,
$\Pi(b_1) = 0.31$,
$\Pi(b_2) = 0.26$
$\Pi(c_1) = 0.07$,
$\Pi(c_2) = 0.31$,
$\Pi(c_3) = 0.26$.
Normalized distributions:
$\Pi_n(a_1) = 0.83$,
$\Pi_n(a_2) = 0.7$,
$\Pi_n(a_3) = 1$,
$\Pi_n(b_1) = 1$,

Uncertain tuples	*Certain tuples*				
N	$a_1b_2c_3$	$a_3b_1c_2$	$a_1b_2c_1$	$a_2b_1c_2$	$a_3b_1c_3$
1	0.81	0.68	0.68	0.68	0.81
2	0.75	0.81	0.68	0.93	0.68

Table 3.11: Similarities between certain and uncertain tuples

Certain tuples	$frequency + similarity$	*maximum projection*
$a_1b_2c_3$	$3 + 0.405$	0.26
$a_3b_1c_2$	4	0.31
$a_1b_2c_1$	1	0.07
$a_2b_1c_2$	$2 + 0.93$	0.22
$a_3b_1c_3$	$1 + 0.405$	0.11

Table 3.12: Affecting uncertain tuples using Dispatching method and computing maximum projection

$\Pi_n(b_2) = 0.83,$
$\Pi_n(c_1) = 0.22,$
$\Pi_n(c_2) = 1,$
$\Pi_n(c_3) = 0.83.$
If we apply max frequency method, we obtain values illustrated by table(3.13):

Now, we compute marginal distributions:
Initial distributions:
$\Pi(a_1) = 0.29,$
$\Pi(a_2) = 0.22,$
$\Pi(a_3) = 0.31,$
$\Pi(b_1) = 0.31,$
$\Pi(b_2) = 0.29,$
$\Pi(c_1) = 0.07,$
$\Pi(c_2) = 0.31,$
$\Pi(c_3) = 0.29.$

Certain tuples	$frequency + similarity$	*maximum projection*
$a_1b_2c_3$	$3 + 0.81$	0.29
$a_3b_1c_2$	4	0.31
$a_1b_2c_1$	1	0.07
$a_2b_1c_2$	$2 + 0.93$	0.22
$a_3b_1c_3$	1	0.07

Table 3.13: Affecting uncertain tuples using Maximum frequency method and computing maximum projection

Normalized distributions:
$\Pi_n(a_1) = 0.93$,
$\Pi_n(a_2) = 0.7$,
$\Pi_n(a_3) = 1$,
$\Pi_n(b_1) = 1$,
$\Pi_n(b_2) = 0.93$
$\Pi_n(c_1) = 0.22$,
$\Pi_n(c_2) = 1$,
$\Pi_n(c_3) = 0.93$

Contrary to Borgelt's method, our approach valorizes more the certain and precise information because it provides high quality information. In our vision there are three characteristics of high quality information: degree of occurrence, precision and certainty.

In any real-life problem, the quality of data is related to ways that datasets are used. That's why we have thought to extract data with high quality and try to bring the data with low quality closer to former ones.

Moreover, our approach can benefit from the simplicity of expressing in possibility theory and treat the two concepts designed by this theory: uncertainty and imprecision.

Our approach can be extended to treat also possibilistic datasets expressing preferences if we use a possibilistic similarity measure used in qualitative

setting because sometimes we encounter several difficulties when expressing exact numerical values and we prefer providing preferences.

3.6 Learning possibilistic networks from possibilistic dataset

After extracting possibility distributions from possibilistic dataset, we can apply one of methods of learning possibilistic networks structure discussed in section(2.3) in the last chapter.

In what follows, we rewrite the greedy parent search algorithm(3.6) but the input in this case is possibility distributions induced using our approach. Note that we will use also evaluation measures listed in the last chapter.

Algorithm 3.6: Greedy parent search

Data: In: local score g, In: Data
begin
  for $i \in \{1, \ldots, n\}$ **do**
  $pa_i \leftarrow \emptyset$
  $g_{old} \leftarrow g(i, pa_i)$
  $Ok \leftarrow true$
  repeat
    $find\ V_j \in Pred(V_i) \setminus pa_i$ **maximizing** $g(i, pa_i \cup V_j)$
    if $g_{new} > g_{old}$ **then**
      $g_{old} \leftarrow g_{new}$
      $pa_i \leftarrow pa_i \cup V_j$
    else
      $ok \leftarrow False$
  until $Ok\ and\ \|pa_i\| < u$;
end

Example 3.11. *Let us consider the possibilistic dataset illustrated by table(3.1) and we apply for example the greedy parent search algorithm with the possibilistic mutual information, expressed by equation(2.11), as local evaluation*

measure. The topological order used by the algorithm is: A,B,C. We then obtain the possibilistic network structure shown in figure(3.2).

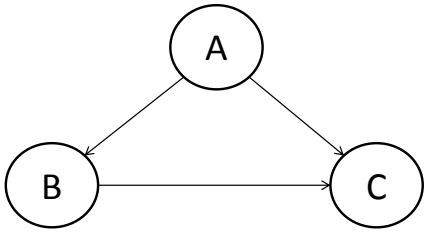

Figure 3.2: The resulted possibilistic network structure

Our approach allows us to learn possibilistic networks structure from possibilistic datasets. Inducing possibilistic distributions has not been well treated before. That's why this area of research has attracted our attention and learning possibilistic networks despite his importance in the reduction of effort and time consumed by experts to express possibility distributions, few works were interested to propose new approaches to perform it.

3.7 Evaluation of learning possibilistic networks structure algorithm

In previous parts, we have discussed several algorithms that have been proposed to learn possibilistic networks structure. In this section, we present a method used to evaluate of learning algorithms. It consists in for a given possibilistic network, we generate a dataset from it. Then, we compare the quality of the initial possibilistic network and the learned one as show in figure(3.3) using a global evaluation measure.

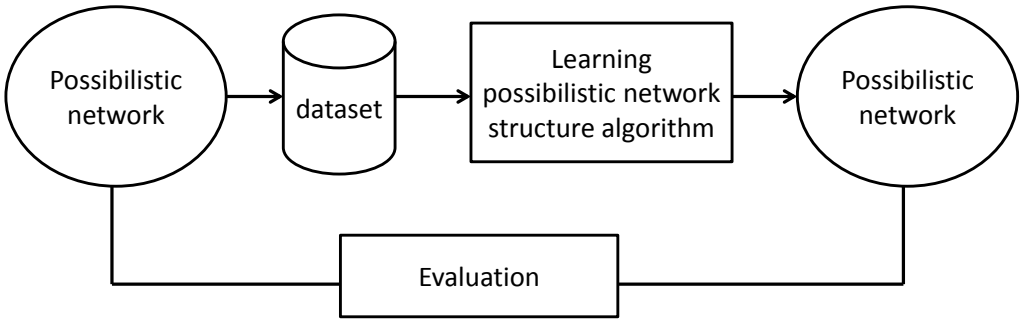

Figure 3.3: Evaluation of learning possibilistic network structure algorithm

3.8 Experimentation

In this section, we present the implementation scheme of our approach. Indeed, we propose a new toolbox named "'Learning Possibilistic Networks"' (LPNT). implemented with Matlab based on the platform BNT. The implementation consists in three parts as shown by figure(3.4). The first is generating a possibilistic dataset. The second one consists in preprocessing the obtained dataset in order to induce possibility distributions. Then we apply greedy parent search algorithm. The last part is devoted to the evaluation of our approach.

3.9 Limitation

It should be noted that despite promising results provided by our approach and the new idea behind it, one particular case can not be efficiently treated which is the extreme case of possibilistic datasets without certain tuples.

To handle this dataset properly, we add another step in preprocessing phase. We can create fictive certain training set from most certain tuples (containing in the majority of attributes precise and certain data). In fact,

	Uncertain training set							
	A			B		C		
N	a_1	a_2	a_3	b_1	b_2	c_1	c_2	c_3
1	0.1	0	1	0.7	1	0.5	1	0.3
2	0	1	0	1	0.8	1	0.6	0.2
3	1	0	0.5	1	0.2	1	0	0
4	1	1	1	1	1	0	0	1
5	0	1	0	1	1	0	1	0
6	1	0	0	1	0	1	1	1
7	0	0	1	0.7	1	1	0	0
8	0.2	0.3	1	1	0	0	1	0

Table 3.14: An example of training set containing only uncertain tuples

we try to create combination of variables very similar to those contained in the dataset and we assign to their frequency 0.

To do so, we take possible combinations of variables in which for each value of their attributes, the degree of possibility is greater than or equal to 0.5.

Example 3.12. *Let us consider the training illustrated by table(3.14) set without any certain tuple: Tuples 5,6,7 and 8 contain in two of three attributes one possible value. So, they can form a fictive certain training set. We can create now (fictive) certain tuples, we extract from the tuple:*

- *5:* $a_2b_1c_2$ *and* $a_2b_2c_2$
- *6:* $a_1b_1c_1$, $a_1b_1c_2$ *and* $a_1b_1c_3$
- *7:* $a_3b_1c_1$ *and* $a_3b_2c_1$
- *8:* $a_3b_1c_2$

Once, we apply our approach normally and without any modification.

3.10 Conclusion

In this chapter, we have proposed a new approach for learning possibilistic networks from possibilistic datasets. The aim of this approach is how can we extract possibility distributions.

This approach offers the possibility to express our opinions in natural way using the simplicity of information presentation in the possibilistic framework.

We have proposed two methods named maximum frequency and dispatching. The two methods require a preprocessing phase. Then, we estimate marginal distributions using a maximum.

Clearly, the two methods can be extended to the qualitative setting of possibility theory. By doing so, our approach will treat all kind of data in the possibilistic framework.

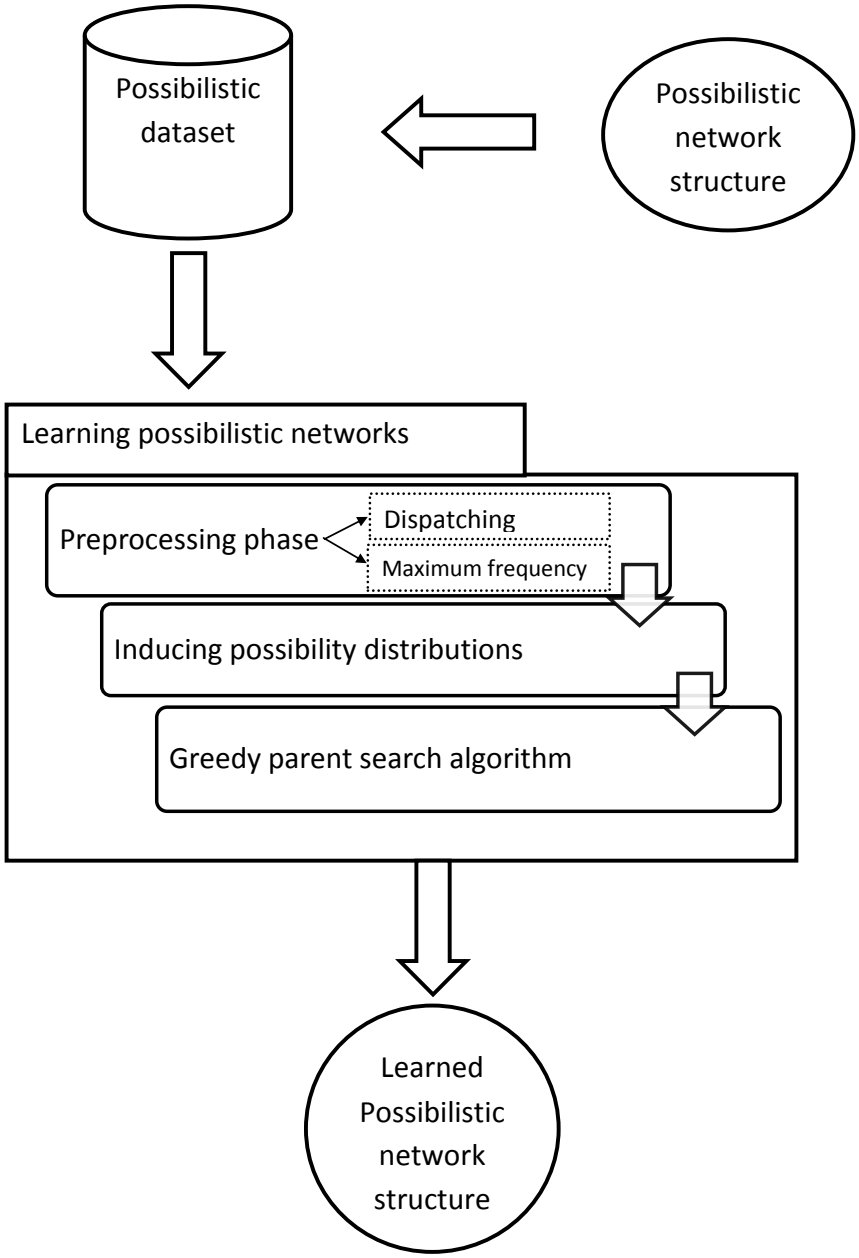

Figure 3.4: Experimentation scheme

Conclusion

In this report, we have proposed a new approach for learning possibilistic networks structure from possibilistic datasets containing attributes described by possibility distributions.

When dealing with possibilistic datasets, the first step we face is how we can estimate marginal possibility distributions. We have proposed an approach which consists in uncertain observations clustering using similarity measures and to handle the case of equal similarities, we have proposed two approaches:

- Dispatching method: It consists in dispatching similarity value to nearest precise and certain observations.
- Maximum frequency method: It consists in affecting the uncertain observation to the most frequent observation among nearest ones.

Once, we can apply one of learning possibilistic networks structure methods using possibility distributions induced with our approach.

As future works, we can first distinguish direct improvements of our proposals. In fact, we aim to adapt proposed methods to deal with possibilistic datasets but in the qualitative setting. Another line of research will be extending possibilistic similarity measures in order to compare possibility distributions in the qualitative setting.

Moreover, it will be useful to enrich this area of research with new evaluation measures that can applied in the possibility theory in its qualitative

setting. Our idea is to propose new methods to learn parameters of possibilistic networks which is the numerical part of them.

Finally, it will be useful to create a toolbox which gathers all methods of learning possibilistic networks structure and parameters with a convivial user-interface.

****

Bibliography

[1] Raouia Ayachi, Nahla Ben Amor, Salem Benferhat, and Rolf Haenni. Compiling possibilistic networks: Alternative approaches to possibilistic inference. In *Proceedings of the Twenty-Sixth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-10)*, pages 40–47, Corvallis, Oregon, 2010. AUAI Press.

[2] Nahla Ben Amor. *Qualitative Possibilistic Graphical Models:From Independence to Propagation Algorithms*. PhD thesis, ISG-University of Tunis, Tunisia, 2002.

[3] Christian Borgelt. *Data Mining with Graphical Models*. PhD thesis, Otto-von-Guericke University Magdeburg, 2000.

[4] Christian Borgelt, Jörg Gebhardt, and Rudolf Kruse. Concepts for probabilistic and possibilistic induction of decision trees on realworld data. In *Proc. 4th European Congress on Intelligent Techniques and Soft Computing (EUFIT'96, Aachen, Germany)*, volume 3, pages 1556–1560, 1996.

[5] Christian Borgelt and Rudolf Kruse. Evaluation measures for learning probabilistic and possibilistic networks. In *Proc. 6th IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE'97, Barcelona, Spain)*, volume 2, pages 1034–1038, 1997.

[6] Christian Borgelt and Rudolf Kruse. Some experimental results on learning probabilistic and possibilistic networks with different evaluation measures. In *Proc. 1st Int. J. Conf. on Qualitative and Quanti-*

tative Practical Reasoning, ECSQARU-FAPR'97, 71–85, pages 71–85. Springer, 1997.

[7] Christian Borgelt and Rudolf Kruse. *Computational intelligence : soft computing and fuzzy-neuro integration with applications*, chapter Probabilistic and Possibilistic Networks and How To Learn Them from Data. Springer, 1998.

[8] Christian Borgelt and Rudolf Kruse. Efficient maximum projection of database-induced multivariate possibility distributions. In *Proc.7th IEEE Int. Conf. on Fuzzy Systems*, 1998.

[9] Christian Borgelt and Rudolf Kruse. Operations and evaluation measures for learning possibilistic graphical models. *Artificial Intelligence*, pages 385–418, August 2003.

[10] Didier Dubois and Henri Prade. *Possibility Theory: An Approach to Computerized Processing of Uncertainty*. Plenum Press, New York, 1988.

[11] Francesco Fabris. On a measure of possibilistic non compatibility with an application to data transmission. *Journal of Interdisciplinary Mathematics*, (3):203–220, October 2002.

[12] Jörg Gebhardt and Rudolf Kruse. A possibilistic interpretation of fuzzy sets by the context model. In *Proc. 1st IEEE Int. Conf. on Fuzzy Systems*, pages 1089–1096, 1992.

[13] Masahiko Higashi and George J. Klir. Fuzzy sets as a basis for a theory of possibility. *International Journal of General Systems*, 9:103–115, 1983.

[14] Ilyes Jenhani. *From Possibilistic Similarity Measures To Possibilistic Decision Trees*. PhD thesis, ISG-University of Tunis, Tunisia, 2010.

[15] Ilyes Jenhani, Nahla Ben Amor, and Zied Elouedi. Information affinity: A new similarity measure for possibilistic uncertain information. In *Proceedings of the 9th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, pages 840–852, 2007.

[16] Ilyes Jenhani, Salem Benferhat, and Zied Elouedi. Properties analysis of inconsistency-based possibilistic similarity measures. In *In Proceedings of the 12th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 173–180, 2008.

[17] Finn Verner Jensen. *Introduction to Bayesian networks.* University College, London: UCL Press, 1996.

[18] Rudolf Kruse, Jörg Gebhardt, and Frank Klawonn. *Foundations of Fuzzy Systems.* wiley, 1994.

[19] Philippe Leray. *Réseaux bayésiens : apprentissage et modélisation de systèmes complexes.* PhD thesis, INSA Rouen, 2006.

[20] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* San Francisco, California: Morgan Kaufmman, 1988.

[21] Ramón Sangüesa and Ulises Cortés. Prior knowledge for learning networks in non-probabilistic settings. *International Journal of Approximate Reasoning*, pages 103–120, April 2000.

[22] Lotfi Askar Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 1:3–28, January 1978.