Probabilistic Graphical Models

Alessandro Antonucci
Cassio P. de Campos
Marco Zaffalon

Technical Report No. IDSIA-01-14
January 2014

IDSIA / USI-SUPSI
Dalle Molle Institute for Artificial Intelligence
Galleria 2, 6928 Manno, Switzerland

Probabilistic Graphical Models

Alessandro Antonucci
Cassio P. de Campos
Marco Zaffalon

January 2014

Abstract

This report[1] presents probabilistic graphical models that are based on imprecise probabilities using a comprehensive language. In particular, the discussion is focused on credal networks and discrete domains. It describes the building blocks of credal networks, algorithms to perform inference, and discusses on complexity results and related work. The goal is to present an easy-to-follow introduction to the topic.

1 Introduction

There is a number of powerful tools for modelling uncertain knowledge with imprecise probabilities. These can be equivalently formalised in terms of coherent sets of desirable gambles, coherent lower previsions, or sets of linear previsions. In the discrete multivariate case, a direct specification of models of this kind might be expensive because of too a high number of joint states, this being exponential in the number of variables. Yet, a compact specification can be achieved if the model displays particular invariance or *composition* properties. The latter is exactly the focus of this chapter: defining a model over its whole set of variables by the composition of a number of sub-models each involving only fewer variables. More specifically, we focus on the kind of composition induced by *independence* relations among the variables. Graphs are particularly suited for the modelling of such independencies, so we formalise our discussion within the framework of probabilistic graphical models. Following these ideas, we introduce a class of probabilistic graphical models with imprecision based on directed graphs called *credal networks*.[2] The example below is used to guide the reader step-by-step through the application of the ideas introduced in this chapter.

Example 1.1 (lung health diagnostic). Assume the lung health status of a patient can be inferred from a probabilistic model over the binary variables: lung cancer (C), bronchitis

[1] This document is a preprint uncorrected chapter from the book *Introduction to Imprecise Probabilities*, Wiley & Sons, 2014.

[2] This chapter mostly discusses credal networks. Motivations for this choice and a short outline on other imprecise probabilistic models is reported in Section 6.

(B), smoker (S), dyspnoea (D), and abnormal X-rays (R).[3] An *imprecise* specification of this model can be equivalently achieved by assessing a coherent set of desirable gambles, a coherent lower prevision or a credal set, all over the joint variable $\mathbf{X} := (C, B, S, D, R)$. This could be demanding because of the exponentially large number of states of the joint variable to be considered (namely, 2^5 in this example). ⧫

Among the different formalisms which can be used to model imprecise probabilistic models, in this chapter we choose credal sets as they appear relatively easy to understand for people used to work with standard (precise) probabilistic models.[4] The next section reports some background information and notation about them.

2 Credal Sets

2.1 Definition and Relation with Lower Previsions

We define a *credal set* (CS) $\mathcal{M}(X)$ over a categorical variable X as a closed convex set of probability mass functions over X.[5] An *extreme point* (or vertex) of a CS is an element of this set which cannot be expressed as a convex combination of other elements. Notation $\mathrm{ext}[\mathcal{M}(X)]$ is used for the set of extreme points of $\mathcal{M}(X)$. We focus on finitely-generated CSs, i.e., sets with a finite number of extreme points. Geometrically speaking, a CS of this kind is a polytope on the probability simplex, which can be equivalently specified in terms of linear constraints to be satisfied by the probabilities of the different outcomes of X (e.g., see Figure 1).[6] As an example, the *vacuous* CS $\mathcal{M}_0(X)$ is defined as the whole set of probability mass functions over X:

$$\mathcal{M}_0(X) := \left\{ P(X) \;\middle|\; \begin{array}{l} P(x) \geq 0, \forall x \in \mathcal{X}, \\ \sum_{x \in \mathcal{X}} P(x) = 1 \end{array} \right\}. \tag{1}$$

The vacuous CS is clearly the largest (and hence least informative) CS we can consider. Any other CS $\mathcal{M}(X)$ over X is defined by imposing additional constraints to $\mathcal{M}_0(X)$.

A single probability mass function $P(X)$ can be regarded as a 'precise' CS made of a single element. Given a real-valued function f of X (which, following the language of the previous chapters, can be also regarded as a *gamble*), its expectation is, in this precise case, $E_P(f) := \sum_{x \in \mathcal{X}} P(x) \cdot f(x)$. This provides a one-to-one correspondence

[3] These variables are referred to the patient under diagnosis and supposed to be self-explanatory. For more insights refer to the *Asia network* [54], which can be regarded as an extension of the model presented here.

[4] There is also a historical motivation for this choice: this chapter is mainly devoted to credal networks with strong independence, which have been described in terms of credal sets from their first formalisation [18].

[5] Previously CSs have been defined as sets of linear previsions instead of probability mass functions. Yet, the one-to-one correspondence between linear previsions and probability mass functions makes the distinction irrelevant. Note also that, in this chapter, we focus on discrete variables. A discussion about extensions to continuous variables is in Section 6.

[6] Standard algorithms can be used to move from the enumeration of the extreme points to the linear constraints generating the CS and *vice versa* (e.g., [8]).

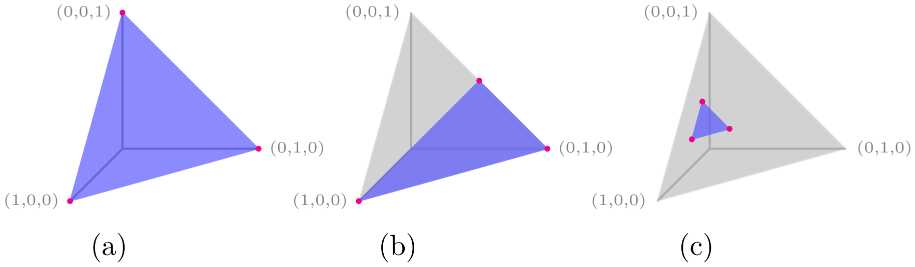

Figure 1: Geometrical representation of CSs over a variable X with $\mathcal{X} = \{x', x'', x'''\}$ in the three-dimensional space with coordinates $[P(x'), P(x''), P(x''')]^T$. Blue polytopes represent respectively: (a) the vacuous CS as in (1); (b) a CS defined by constraint $P(x''') \geq P(x'')$; (c) a CS $\mathcal{M}(X)$ such that $\mathrm{ext}[\mathcal{M}(X)] = \{[.1, .3, .6]^T, [.3, .3, .4]^T, [.1, .5, .4]^T\}$. The extreme points are in magenta.

between probability mass functions and *linear* previsions. Given a generic CS $\mathcal{M}(X)$, we can evaluate the lower expectation $\underline{E}_{\mathcal{M}}(f) := \min_{P(X)\in\mathcal{M}(X)} P(f)$ (and similarly for the upper). This defines a coherent lower prevision as a lower envelope of a set of linear previsions. As an example, the vacuous CS $\mathcal{M}_0(X)$ in (1) defines the (vacuous) coherent lower prevision, i.e., $\underline{E}_{\mathcal{M}_0}(f) = \min_{x\in\mathcal{X}} f(x)$. Note that a set and its convex closure have the same lower envelope, and hence a set of distributions and its convex closure define the same coherent lower prevision. This means that, when computing expectations, there is no lack of generality in defining CSs only as closed convex sets of probability mass functions, and the correspondence with coherent lower previsions is bijective [67, Section 3.6.1]. Note also that the optimization task associated to the above definition of $\underline{E}_{\mathcal{M}}(f)$ is an LP optimization problem, whose solution can be equivalently obtained by considering only the extreme points of the CS [25], i.e.,

$$\underline{E}_{\mathcal{M}}(f) = \min_{P(X)\in\mathrm{ext}[\mathcal{M}(X)]} E_P(f). \tag{2}$$

The above discussion also describes how inference with CSs is intended. Note that the complexity of computations as in (2) is linear in the number of extreme points, this number being unbounded for general CSs.[7]

A notable exception is the Boolean case: a CS over a binary variable[8] cannot have more than two extreme points. This simply follows from the fact that the probability simplex (i.e., the vacuous CS) is a one-dimensional object.[9] As a consequence of that, any CS over a binary variable can be specified by simply requiring the probability of a single outcome

[7] Some special classes of CSs with bounded number of extreme points are the vacuous ones as in (1) and those corresponding to linear-vacuous mixtures (for which the number of the extreme points cannot exceed the cardinality of $\mathcal{X}$). Yet, these theoretical bounds are not particularly binding for (joint) variables with high dimensionality.

[8] If X is a binary variable, its two states are denoted by x and $\neg x$.

[9] Convex sets on one-dimensional varieties are isomorphic to intervals on the real axis, whose extreme points are the lower and upper bounds.

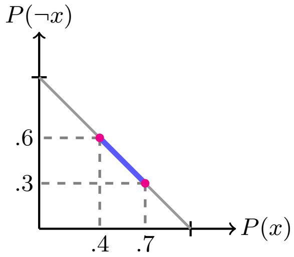

Figure 2: Geometrical representation (in blue) of a CS over a binary variable X in the two-dimensional space with coordinates $[P(x), P(\neg x)]^T$. The two extreme points are in magenta, while the probability simplex, which corresponds to the vacuous CS, is in grey.

to belong to an interval. E.g., if $\mathcal{M}(X) := \{P(X) \in \mathcal{M}_0(X) | .4 \leq P(X = x) \leq .7\}$, then $\text{ext}[\mathcal{M}(X)] = \{[.4, .6]^T, [.7, .3]^T\}$ (see also Figure 2).

2.2 Marginalisation and Conditioning

Given a joint CS $\mathcal{M}(X, Y)$, the corresponding *marginal* CS $\mathcal{M}(X)$ contains all the probability mass functions $P(X)$ which are obtained by marginalising out Y from $P(X, Y)$, for each $P(X, Y) \in \mathcal{M}(X, Y)$. Notably, the marginal CS can be equivalently obtained by only considering its extreme points, i.e.,

$$\mathcal{M}(X) = \text{CH}\left\{ P(X) \;\middle|\; \begin{array}{l} P(x) := \sum_{y \in \mathcal{Y}} P(x, y), \forall x \in \mathcal{X}, \\ \forall P(X, Y) \in \text{ext}[\mathcal{M}(X, Y)] \end{array} \right\}, \tag{3}$$

where CH denotes the convex hull operation.[10]

We similarly proceed for *conditioning*. For each $y \in \mathcal{Y}$, the conditional CS $\mathcal{M}(X|y)$ is made of all the conditional mass functions $P(X|y)$ obtained from $P(X, Y)$ by Bayes rule, for each $P(X, Y) \in \mathcal{M}(X, Y)$ (this can be done under the assumption $P(y) > 0$ for each mass function $P(X, Y) \in \mathcal{M}(X, Y)$, i.e., $\underline{P}(y) > 0$). As in the case of marginalisation, conditional CSs can be obtained by only considering the extreme points of the joint CS, i.e.,

$$\mathcal{M}(X|y) = \text{CH}\left\{ P(X|y) \;\middle|\; \begin{array}{l} P(x|y) := \frac{P(x,y)}{\sum_{x \in \mathcal{X}} P(x,y)}, \forall x \in \mathcal{X}, \\ \forall P(X, Y) \in \text{ext}[\mathcal{M}(X, Y)] \end{array} \right\}. \tag{4}$$

The following notation is used as a shortcut for the collection of conditional CSs associated to all the possible values of the conditioning variable: $\mathcal{M}(X|Y) := \{\mathcal{M}(X|y)\}_{y \in \mathcal{Y}}$.

Example 2.1. In the medical diagnosis setup of Example 1.1, consider only variables lung cancer (C) and smoker (S). The available knowledge about the joint states of these two

[10] In order to prove that the CS in (3) is consistent with the definition of marginal CS, it is sufficient to check that any extreme point of $\mathcal{M}(X)$ is obtained marginalizing out Y from an extreme point of $\mathcal{M}(X, Y)$. If that would not be true, we could express an extreme point of $\mathcal{M}(X)$ as the marginalization of a convex combination of two or more extreme points of $\mathcal{M}(X, Y)$, and hence as a convex combination of two or more probability mass functions over X. This is against the original assumptions.

variables is modelled by a CS $\mathcal{M}(C,S) = \mathrm{CH}\{P_j(C,S)\}_{j=1}^{8}$, whose eight extreme points are those reported in Table 1 and depicted in Figure 3. It is indeed straightforward to compute the marginal CS for variable S as in (3):

$$\mathcal{M}(S) = \mathrm{CH}\left\{ \left[\begin{array}{c} \frac{1}{4} \\ \frac{3}{4} \end{array} \right], \left[\begin{array}{c} \frac{5}{8} \\ \frac{3}{8} \end{array} \right] \right\}. \tag{5}$$

Similarly, the conditional CSs for variable C as in (4) given the two values of S are:

$$\mathcal{M}(C|s) = \mathrm{CH}\left\{ \left[\begin{array}{c} \frac{1}{4} \\ \frac{3}{4} \end{array} \right], \left[\begin{array}{c} \frac{3}{4} \\ \frac{1}{4} \end{array} \right] \right\} \quad , \quad \mathcal{M}(C|\neg s) = \mathrm{CH}\left\{ \left[\begin{array}{c} \frac{1}{7} \\ \frac{6}{7} \end{array} \right], \left[\begin{array}{c} \frac{3}{4} \\ \frac{1}{4} \end{array} \right] \right\}. \tag{6}$$

♦

j	1	2	3	4	5	6	7	8
$P_j(c,s)$	$\frac{1}{8}$	$\frac{1}{4}$	$\frac{3}{8}$	$\frac{3}{16}$	$\frac{3}{8}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{3}{16}$
$P_j(\neg c,s)$	$\frac{1}{8}$	$\frac{3}{8}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{3}{8}$	$\frac{3}{8}$
$P_j(c,\neg s)$	$\frac{9}{16}$	$\frac{1}{4}$	$\frac{3}{8}$	$\frac{9}{16}$	$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$
$P_j(\neg c,\neg s)$	$\frac{3}{16}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{3}{16}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{3}{8}$	$\frac{3}{8}$

Table 1: The eight extreme points of the joint CS $\mathcal{M}(C,S) = \mathrm{CH}\{P_j(C,S)\}_{j=1}^{8}$. Linear algebra techniques (e.g., see [8], even for a software implementation) can be used to check that none of these distributions belong to the convex hull of the remaining seven.

2.3 Composition

Let us define a *composition* operator in the imprecise-probabilistic framework. Given a collection of conditional CSs $\mathcal{M}(X|Y)$ and a marginal CS $\mathcal{M}(Y)$, the marginal extension introduced in Chapter within the language of coherent lower previsions, corresponds to the following specification of a joint CS as a composition of $\mathcal{M}(Y)$ and $\mathcal{M}(X|Y)$:

$$\mathcal{M}(X,Y) := \mathrm{CH}\left\{ P(X,Y) \left| P(x,y) := P(x|y) \cdot P(y) \begin{array}{l} \forall x \in \mathcal{X}, \forall y \in \mathcal{Y}, \\ \forall P(Y) \in \mathcal{M}(Y), \\ \forall P(X|y) \in \mathcal{M}(X|y) \end{array} \right. \right\}. \tag{7}$$

Notation $\mathcal{M}(X|Y) \otimes \mathcal{M}(Y)$ will be used in the following as a shortcut for the right-hand side of (7). As usual, the joint CS in (7) can be equivalently obtained by considering only the extreme points, i.e.,

$$\mathcal{M}(X|Y) \otimes \mathcal{M}(Y) = \mathrm{CH}\left\{ P(X,Y) \left| P(x,y) := P(x|y) \cdot P(y) \begin{array}{l} \forall x \in \mathcal{X}, \forall y \in \mathcal{Y}, \\ \forall P(Y) \in \mathrm{ext}[\mathcal{M}(Y)], \\ \forall P(X|y) \in \mathrm{ext}[\mathcal{M}(X|y)] \end{array} \right. \right\}. \tag{8}$$

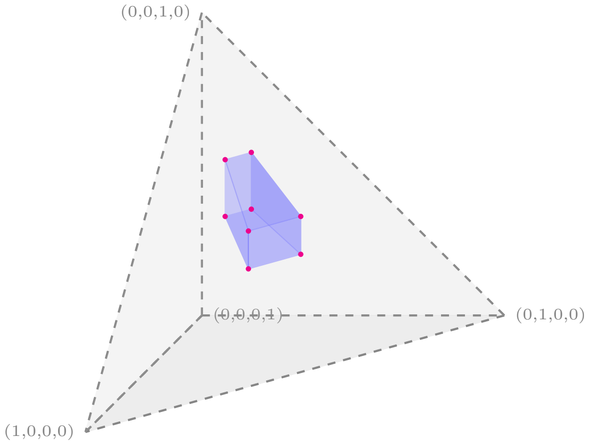

Figure 3: Geometrical representation of the CS over a (joint) quaternary variable (C, S), with both C and S binary, as quantified in Table 1. The CS is the blue polyhedron (and its eight extreme points are in magenta), while the probability simplex is the grey tetrahedron. The representation is in the three-dimensional space with coordinates $[P(c, s), P(\neg c, s), P(c, \neg s)]^T$, which are the barycentric coordinates of the four-dimensional probability simplex.

Example 2.2. As an exercise, compute by means of (8) the composition of $\mathcal{M}(C|S) \otimes \mathcal{M}(S)$ of the unconditional CS in (5) and the conditional CSs in (6). In this particular case, the so obtained CS $\mathcal{M}(C, S)$ coincides with that as in Table (1). ♦

Example 2.3. As a consequence of (7), we may define a joint CS over the variables in Example 1.1 by means of the following composition:

$$\mathcal{M}(D, R, B, C, S) = \mathcal{M}(D|R, B, C, S) \otimes \mathcal{M}(R, B, C, S),$$

and then, iterating, [11]

$$\mathcal{M}(D, R, B, C, S) = \mathcal{M}(D|R, B, C, S) \otimes \mathcal{M}(R|B, C, S) \otimes \mathcal{M}(B|C, S) \otimes \mathcal{M}(C|S) \otimes \mathcal{M}(S) \tag{9}$$

♦

Note that (9) does not make the specification of the joint CS less demanding (the number of probabilistic assessments we should make for the CS on the left-hand side is almost the same required by the first on the right-hand side). In next section, we show how independence can make the specification of these multivariate models more compact.

[11] Brackets setting the composition ordering in (9) are omitted because of the associativity of the composition operator $\otimes$.

3 Independence

First, let us formalise the notion of independence in the precise probabilistic framework. Consider variables X and Y, and assume that a (precise) joint probability mass function $P(X,Y)$ models the knowledge about their joint configurations. We say that X and Y are *stochastically* independent if $P(x,y) = P(x) \cdot P(y)$ for each $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, where $P(X)$ and $P(Y)$ are obtained from $P(X,Y)$ by marginalisation.

The concept might be easily extended to the imprecise probabilistic framework by the notion of *strong independence*.[12] Given a joint CS $\mathcal{M}(X,Y)$, X and Y are *strongly* independent if, for all $P(X,Y) \in \text{ext}[\mathcal{M}(X,Y)]$, X and Y are stochastically independent, i.e., $P(x,y) = P(x) \cdot P(y)$, for each $x \in \mathcal{X}$, $y \in \mathcal{Y}$. The concept admits also a formulation in the conditional case. Variables X and Y are strongly independent given Z if, for each $z \in \mathcal{Z}$, every $P(X,Y|z) \in \text{ext}[K(X,Y|z)]$ factorises as $P(x,y|z) = P(x|z) \cdot P(y|z)$, for each $x \in \mathcal{X}$ and $y \in \mathcal{Y}$.

Example 3.1. In Example 1.1, consider only variables C, B and S. According to (8):

$$\mathcal{M}(C,B,S) = \mathcal{M}(C,B|S) \otimes \mathcal{M}(S).$$

Assume that, once you know whether or not the patient is a smoker, there is no relation between the fact that he could have lung cancer and bronchitis. This can be regarded as a conditional independence statement regarding C and B given S. In particular, we consider the notion of strong independence as above. This implies the factorisation $P(c,b|s) = P(c|s) \cdot P(b|s)$ for each possible value of the variables and each extreme point of the relative CSs. Expressing that as a composition, we have:[13]

$$\mathcal{M}(C,B,S) = \mathcal{M}(C|S) \otimes \mathcal{M}(B|S) \otimes \mathcal{M}(S). \tag{10}$$

♦

In this particular example, the composition in (10) is not providing a significantly more compact specification of the joint CS $\mathcal{M}(C,B,S)$. Yet, for models with more variables and more independence relations, this kind of approach leads to a substantial reduction of the number of states to be considered for the specification of a joint model. In the rest of this section, we generalise these ideas to more complex situations where a number of conditional independence assessments is provided over a possibly large number of variables. In order to do that, we need a compact language to describe conditional independence among variables. This is typically achieved in the framework of probabilistic graphical models, by assuming a one-to-one correspondence between the variables under

[12] Strong independence is not the only independence concept proposed within the imprecise-probabilistic framework. See Section 6 for pointers on imprecise probabilistic graphical models based on other concepts.

[13] In (10) the composition operator has been extended to settings more general than (8). With marginal CSs, a joint CS $\mathcal{M}(X,Y) := \mathcal{M}(X) \otimes \mathcal{M}(Y)$ can be obtained by taking all the possible combinations of the extreme points of the marginal CSs (and then taking the convex hull). Thus, $\mathcal{M}(X,Y|Z) := \mathcal{M}(X|Z) \otimes \mathcal{M}(Y|Z)$ is just the conditional version of the same relation. Similarly, $\mathcal{M}(X,Z|Y,W) := \mathcal{M}(X|Y) \otimes \mathcal{M}(Z|W)$. Notably, even in these extended settings, the composition operator remains associative.

consideration and the nodes of a directed acyclic[14] graph and then by assuming the so-called strong *Markov condition*:[15]

any variable is strongly independent
of its non-descendants non-parents given its parents.

We point the reader to [27] for an axiomatic approach to the modelling of probabilistic independence concepts by means of directed (and undirected) graphs. Here, in order to clarify the semantics of this condition, we consider the following example.

Example 3.2. Assume a one-to-one correspondence between the five binary variables in Example 1.1 and the nodes of the directed acyclic graph in Figure 4. The strong Markov condition for this graph implies the following conditional independence statements:

- given smoker, lung cancer and bronchitis are strongly independent;
- given smoker, bronchitis and abnormal X-rays are strongly independent;
- given lung cancer, abnormal X-rays and dyspnoea are strongly independent, and abnormal X-rays and smoker are strongly independent;
- given lung cancer and bronchitis, dyspnoea and smoker are strongly independent.

The above independence statements can be used to generate further independencies by means of the axioms in [27]. ⧫

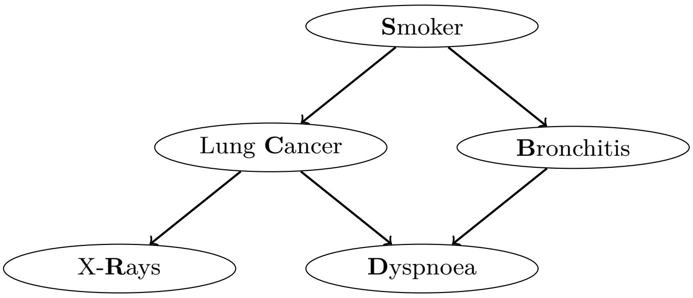

Figure 4: A directed graph over the variables in Example 1.1.

4 Credal Networks

Let us introduce the definition of credal network by means of the following example.

[14] A cycle in a directed graph is a directed path connecting a node with itself. A directed graph is acyclic if no cycles are present in it.

[15] Assuming a one-to-one correspondence between a set of variables and the nodes of a directed graph, the *parents* of a variable are the variables corresponding to the immediate predecessors. Analogously, we define the *children* and, by iteration, the *descendants* of a node/variable.

Example 4.1. Consider the variables in Example 1.1 associated to the graph in Figure 4. Assume that, for each variable, conditional CSs given any possible value of the parents have been assessed. This means that $\mathcal{M}(S)$, $\mathcal{M}(C|S)$, $\mathcal{M}(B|S)$, $\mathcal{M}(R|C)$, and $\mathcal{M}(D|C,B)$ are available. A joint CS can be defined by means of the following composition:

$$\mathcal{M}(D,R,B,C,S) := \mathcal{M}(D|C,B) \otimes \mathcal{M}(R|C) \otimes \mathcal{M}(B|S) \otimes \mathcal{M}(C|S) \otimes \mathcal{M}(S). \tag{11}$$

♦

In general situations, we aim at specifying a probabilistic graphical model over a collection of categorical variables $\mathbf{X} := (X_1, \ldots, X_n)$, which are in one-to-one correspondence with the nodes of a directed acyclic graph $\mathcal{G}$. The notation $\mathrm{Pa}(X_i)$ is used for the variables corresponding to the parents of X_i according to the graph $\mathcal{G}$ (e.g., in Figure 4, $\mathrm{Pa}(D) = (C, B)$). Similarly, $\mathrm{pa}(X_i)$ and $Pa(X_i)$ are the generic value and possibility space of $\mathrm{Pa}(X_i)$. Assume the variables in $\mathbf{X}$ to be in a topological ordering.[16] Then, by analogy with what we did in Example 4.1, we can define a joint CS as follows:

$$\mathcal{M}(\mathbf{X}) := \otimes_{i=n,\ldots,1} \mathcal{M}(X_i|\mathrm{Pa}(X_i)). \tag{12}$$

This leads to the following.

Definition 4.2. A credal network (CN) over a set of variables $\mathbf{X} := (X_1, \ldots, X_n)$ is a pair $\langle \mathcal{G}, \mathbb{M} \rangle$, where $\mathcal{G}$ is a directed acyclic graph whose nodes are associated to $\mathbf{X}$, and $\mathbb{M}$ is a collection of conditional CSs $\{\mathcal{M}(X_i|\mathrm{Pa}(X_i))\}_{i=1,\ldots n}$, where $\mathcal{M}(X_i|\mathrm{Pa}(X_i)) = \{\mathcal{M}(X_i|\mathrm{pa}(X_i))\}_{\mathrm{pa}(X_i) \in Pa(X_i)}$. The joint CS $\mathcal{M}(\mathbf{X})$ in (12) is called the strong extension of the CN.

A characterization of the extreme points of the strong extension $\mathcal{M}(\mathbf{X})$ as in (12) is provided by the following proposition [6].

Proposition 4.3. *Let $\{P_j(\mathbf{X})\}_{j=1}^{v}$ denote the extreme points of the strong extension $\mathcal{M}(\mathbf{X})$ of a CN, i.e., $\mathrm{ext}[\mathcal{M}(\mathbf{X})] = \{P_j(\mathbf{X})\}_{j=1}^{v}$. Then, for each $j = 1, \ldots, v$, $P_j(\mathbf{X})$ is a joint mass functions obtained as the product of extreme points of the conditional CSs, i.e., $\forall \mathbf{x} \in \mathcal{X}$:*

$$P_j(\mathbf{x}) = \prod_{i=1}^{n} P_j(x_i|\mathrm{pa}(X_i)), \tag{13}$$

where, for each $i = 1, \ldots, n$ and $\mathrm{pa}(X_i) \in Pa(X_i)$, $P_j(X_i|\mathrm{pa}(X_i)) \in \mathrm{ext}[\mathcal{M}(X_i|\mathrm{pa}(X_i))]$.

According to Proposition 4.3, the extreme points of the strong extension of a CN can be obtained by combining the extreme points of the conditional CSs involved in its specification. Note that this can make the number of extreme points of the strong extension exponential in the input size.

[16] A topological ordering for the nodes of a directed acyclic graph is an ordering in which each node comes before all nodes to which it has outbound arcs. As an example, (S, C, B, R, D) is a topological ordering for the nodes of the graph in Figure 4. Note that every directed acyclic graph has one or more topological orderings.

Example 4.4. The CS in (11) can be regarded as the strong extension of a CN. According to Proposition 4.3, each vertex of it factorises as follows:

$$P(d,r,b,c,s) = P(d|c,b)P(r|c)P(b|s)P(c|s)P(s).$$

It is a simple exercise to verify that this joint distribution satisfies the conditional independence statements following from the Markov condition (indented with the notion of stochastic instead of strong independence). ♦

The above result can be easily generalised to the strong extension of any CN. Thus, if any extreme point of the strong extension obeys the Markov condition with stochastic independence, the strong extension satisfies the Markov condition with strong independence.

An example of CN specification and its strong extension are reported in the following.

Example 4.5. Given the five binary variables introduced in Example 1.1, associated to the directed acyclic graph in Figure 4, consider the following specification of the (collections of conditional) CSs $\mathcal{M}(S)$, $\mathcal{M}(C|S)$, $\mathcal{M}(B|S)$, $\mathcal{M}(R|C)$, $\mathcal{M}(D|C,B)$, implicitly defined by the following constraints:

$$\begin{cases} .25 \leq P(s) \leq .50 \\ \\ .05 \leq P(c|\neg s) \leq .10 \\ .15 \leq P(c|s) \leq .40 \\ \\ .20 \leq P(b|\neg s) \leq .30 \\ .30 \leq P(b|s) \leq .55 \\ \\ .01 \leq P(r|\neg c) \leq .05 \\ .90 \leq P(r|c) \leq .99 \\ \\ .10 \leq P(d|\neg c, \neg b) \leq .20 \\ .80 \leq P(d|\neg c, b) \leq .90 \\ .60 \leq P(d|c, \neg b) \leq .80 \\ .90 \leq P(d|c, b) \leq .99. \end{cases}$$

The strong extension of this CN is a CS $\mathcal{M}(D,R,B,C,S)$ defined as in (12). As a consequence of Proposition 4.3, the extreme points of $\mathcal{M}(D,R,B,C,S)$ are combinations of the extreme points of the local CSs, and can therefore be up to 2^{11} if no combination lies in the convex hull of the others. As a simple exercise let us compute the lower probability for the joint state where all the variables are in the state true, i.e., $\underline{P}(d,r,b,c,s)$. This lower probability should be intended as the minimum, with respect to the strong extension $\mathcal{M}(D,R,B,C,S)$, of the joint probability $P(d,r,b,c,s)$. Thus, we have:

$$\min_{P(D,R,B,C,S)\in\mathcal{M}(D,R,B,C,S)} P(d,r,b,c,s) = \min_{P(D,R,B,C,S)\in\mathrm{ext}[\mathcal{M}(D,R,B,C,S)]} P(d,r,b,c,s)$$

$$= \min_{\substack{P(S) \in \text{ext}[\mathcal{M}(S)] \\ P(C|s) \in \text{ext}[\mathcal{M}(C|s)] \\ P(B|s) \in \text{ext}[\mathcal{M}(B|s)] \\ P(R|c) \in \text{ext}[\mathcal{M}(R|c)] \\ P(D|c) \in \text{ext}[\mathcal{M}(D|c,s)]}} P(s)P(c|s)P(b|s)P(r|c)P(d|c,s) = \underline{P}(s)\underline{P}(c|s)\underline{P}(b|s)\underline{P}(r|c)\underline{P}(d|c,s),$$

with the first step because of (2), the second because of Proposition 4.3, and the last because each conditional distribution take its values independently of the others. This result, together with analogous for upper probability, gives $P(d, r, b, c, s) \in [.0091125, .1078110]$. ♦

The above computation can be regarded as a simple example of inference based on the strong extension of a CN. More challenging problems based on more sophisticated algorithmic techniques are described in Section 5.3.

Overall, we introduced CNs as a well-defined class of probabilistic graphical models with imprecision. Note that exactly as a single probability mass function can be regarded as a special CS with a single extreme point, we can consider a special class of CNs, whose conditional CSs are made of a single probability mass function each. This kind of CNs are called *Bayesian networks* [60] and their strong extension is a single joint probability mass function, which factorises according to the (stochastic) conditional independence relations depicted by its graph, i.e., as in (13). In this sense, CNs can be regarded as a generalisation to imprecise probabilities of Bayesian networks. With respect to these precise probabilistic graphical models, CNs should be regarded as a more expressive class of models.

4.1 Non-Separately Specified Credal Networks

In the definition of strong extension as in (12), each conditional probability mass function is free to vary in (the set of extreme points of) its conditional CS independently of the others. In order to emphasize this feature, CNs of this kind are said to be defined with separately specified CSs, or simply *separately specified.* Separately specified CNs are the most commonly used type of CN, but it is possible to consider CNs whose strong extension cannot be formulated as in (12). This corresponds to having relationships between the different specifications of the conditional CSs, which means that the choice for a given conditional mass function can be affected by that of some other conditional mass functions. A CN of this kind is simply called *non-separately specified.*

As an example, some authors considered so-called *extensive* specifications, where instead of a separate specification for each conditional mass function associated to X_i, the *probability table* $P(X_i|\text{Pa}(X_i))$, i.e., a function of both X_i and $\text{Pa}(X_i)$, is defined to belong to a finite set (of tables). This corresponds to assuming constraints between the specification of the conditional CSs $\mathcal{M}(X_i|\text{pa}(X_i))$ corresponding to the different values of $\text{pa}(X_i) \in Pa(X_i)$. The strong extension of an extensive CN is obtained as in (12), by simply replacing the separate requirements for each single conditional mass function with extensive requirements about the tables which take values in the corresponding finite set.

Example 4.6 (extensive specification)*.* Consider the CN defined in Example 4.5 over the graph in Figure 4. Keep the same specification of the conditional CSs, but this time use extensive constraints for the CSs of B. According to Definition 4.2, in the joint specifications of the two CSs of B, all the four possible combinations of the extreme points of $\mathcal{M}(B|s)$ with those of $\mathcal{M}(B|\neg s)$ appear. An example of extensive specification for this variable would imply that only the following two tables can be considered:

$$P(B|S) \in \left\{ \left[\begin{array}{c|c} .20 & .30 \\ .80 & .70 \end{array} \right], \left[\begin{array}{c|c} .30 & .55 \\ .70 & .45 \end{array} \right] \right\}. \tag{14}$$

♦

Extensive specifications are not the only kind of non-separate specification we consider for CNs. In fact, we can also consider constraints between the specification of conditional CSs corresponding to different variables. This is a typical situation when the quantification of the conditional CSs in a CN is obtained from a data set. A simple example is illustrated below.

Example 4.7 (learning from incomplete data)*.* Among the five variables of Example 1.1, consider only S, C, and R. Following Example 3.2, S and R are strongly independent given C. Accordingly, let us define a joint $\mathcal{M}(S, C, R)$ as a CN associated to a graph corresponding to a chain of three nodes, with S first (parentless) node and R last (childless) node of the chain. Assume that we learn the model probabilities from the incomplete data set in Table 2, assuming no information about the process making the observation of C missing in the last instance of the data set. A possible approach is to learn two distinct probabilities from the two complete data set corresponding to the possible values of the missing observation, and use them to specify the extreme points of the conditional CSs of a CN.

S	C	R
s	c	r
$\neg s$	$\neg c$	r
s	c	$\neg r$
s	$*$	r

Table 2: A data set about three of the five binary variables of Example 1.1; '$*$' denotes a missing observation.

To make things simple we compute the probabilities for the joint states by means of the relative frequencies in the complete data sets. Let $P_1(S, C, R)$ and $P_2(S, C, R)$ be the joint mass functions obtained in this way, from which we obtain the same conditional mass functions for

$$\begin{aligned} P_1(s) &= P_2(s) = \tfrac{3}{4} \\ P_1(c|\neg s) &= P_2(c|\neg s) = 0 \\ P_1(r|\neg c) &= P_2(r|\neg c) = 1; \end{aligned}$$

and different conditional mass functions for

$$\begin{array}{ll} P_1(c|s) = 1 & P_2(c|s) = \tfrac{2}{3} \\ P_1(r|c) = \tfrac{2}{3} & P_2(r|c) = \tfrac{1}{2}. \end{array} \tag{15}$$

We have therefore obtained two, partially distinct, specifications for the local models over variables S, C and R. The conditional probability mass functions of these networks are the extreme points of the conditional CSs for the CN we consider. Such a CN is non-separately specified. To see that, just note that if the CN would be separately specified the values $P(c|s) = 1$ and $P(r|c) = \frac{1}{2}$ could be regarded as a possible instantiation of the conditional probabilities, despite the fact that there are no complete data sets leading to this combination of values. ⧫

Although their importance in modelling different problems, non-separate CNs have received relatively small attention in the literature. Most of the algorithms for CN inference are in fact designed for separately specified CNs. However, two important exceptions are two credal classifiers which are presented later: the naive credal classifier and the credal TAN. Furthermore, it has been shown that non-separate CNs can be equivalently described as separately specified CNs augmented by a number of auxiliary parent nodes enumerating only the possible combinations for the constrained specifications of the conditional CSs. This can be described by the following example.

Example 4.8 ('separating' a non-separately specified CN)*.* Consider the extensively specified CN in Example 4.6. Augment this network with an auxiliary node A, which is used to model the constraints between the two, non-separately specified, conditional CSs $\mathcal{M}(B|s)$ and $\mathcal{M}(B|\neg s)$. Node A is therefore defined as a parent of B, and the resulting graph becomes that in Figure 5. The states of A are indexing the possible specifications of the table $P(B|S)$. So, A should be a binary variable such that $P(B|S, a)$ and $P(B|S, \neg a)$ are the two tables in (14). Finally, specify $\mathcal{M}(A)$ as a vacuous CS. Overall, we obtain a separately specified CN whose strong extension coincides with that of the CN in Example 4.6.[17] ⧫

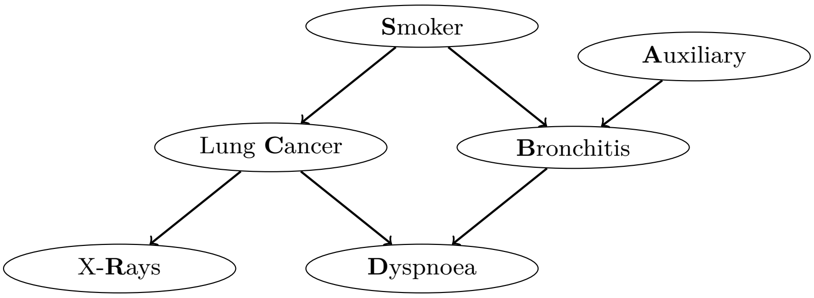

Figure 5: The network in Figure 4 with an auxiliary node indexing the tables providing the extensive specification of $\mathcal{M}(B|S)$.

This procedure can be easily applied to any non-separate specification of a CN. We point the reader to [6] for details.

[17] Once rather than the auxiliary variable A is marginalized out.

5 Computing with Credal Networks

5.1 Credal Networks Updating

In the previous sections we have shown how a CN can model imprecise knowledge over a joint set of variables. Once this modelling phase has been achieved, it is possible to interact with the model through *inference* algorithms. This corresponds, for instance, to query a CN in order to gather probabilistic information about a variable of interest X_q given evidence x_E about some other variables X_E. This task is called *updating* and consists in the computation of the lower (and upper) posterior probability $P(x_q|x_E)$ with respect to the network strong extension $\mathcal{M}(\mathbf{X})$. For this specific problem, (2) rewrites as follows:

$$\underline{P}(x_q|x_E) = \min_{P(\mathbf{X})\in\mathcal{M}(\mathbf{X})} P(x_q|x_E) = \min_{j=1,\dots,v} \frac{\sum_{x_M}\prod_{i=1}^{n} P_j(x_i|\mathrm{pa}(X_i))}{\sum_{x_M,x_q}\prod_{i=1}^{n} P_j(x_i|\mathrm{pa}(X_i))}, \tag{16}$$

where $\{P_j(\mathbf{X})\}_{j=1}^{v}$ are the extreme points of the strong extension, $X_M = \mathbf{X}\setminus(\{X_q\}\cup X_E)$, and in the second step we exploit the result in Proposition 4.3. A similar expression with a maximum replacing the minimum defines upper probabilities $\overline{P}(x_q|x_E)$. Note that, for each $j = 1,\dots,v$, $P_j(x_q|x_E)$ is a posterior probability for a Bayesian network over the same graph. In principle, updating could be therefore solved by simply iterating standard Bayesian network algorithms. Yet, according to Proposition 4.3, the number v of extreme points of the strong extension might be exponential in the input size, and (16) can be hardly solved by such an exhaustive approach. In fact, exact updating displays higher complexity in CNs rather than Bayesian networks: CNs updating is NP-complete for polytrees[18] (while polynomial-time algorithms exist for Bayesian networks with the same topology [60]), and $\mathrm{NP}^{\mathrm{PP}}$-complete for general CNs [30] (while updating of general Bayesian networks is PP-complete [57]). Yet, a number of exact and approximate algorithm for CNs updating has been developed. A summary about the state of the art in this field is reported in Section 5.3.

Algorithms of this kind can compute, given the available evidence x_E, the lower and upper probabilities for the different outcomes of the queried variable X_q, i.e., the set of probability intervals $\{[\underline{P}(x_q|x_E),\overline{P}(x_q|x_E)]\}_{x_q\in\mathcal{X}_q}$. In order to identify the most probable outcome for X_q, a simple *interval dominance* criterion can be adopted. The idea is to *reject* a value of X_q if its upper probability is smaller than the lower probability of some other outcome. Clearly, this criterion is not always intended to return a single value as the most probable for X_q. In general, after updating, the posterior knowledge about the state of X_q is described by the set $\mathcal{X}_q^* \subseteq \mathcal{X}_q$, defined as follows:

$$\mathcal{X}_q^* := \left\{ x_q \in \mathcal{X}_q \,\middle|\, \nexists\, x_q' \in \mathcal{X}_q \; s.t.\, \overline{P}(x_q|x_E) < \underline{P}(x_q'|x_E) \right\}. \tag{17}$$

Criteria other than interval dominance have been proposed in the literature and formalised in the more general framework of decision making with imprecise probabilities.

[18] A credal (or a Bayesian) network is said to be a *polytree* if its underlying graph is singly connected, i.e., if given two nodes there is at most a single undirected path connecting them. A *tree* is a polytree whose nodes cannot have more than a single parent.

As an example, the set of non-dominated outcomes $\mathcal{X}_q^{**}$ according to the *maximality* criterion [67, Section 3.9] is obtained by rejecting the outcomes whose probabilities are dominated by those of some other outcome, for any distribution in the posterior CS, i.e.,

$$\mathcal{X}_q^{**} := \left\{ x_q \in \mathcal{X}_q \,\middle|\, \nexists x'_q \in \mathcal{X}_q \,\text{s.t.}\, P(x_q|x_E) < P(x'_q|x_E) \forall P(X_q|x_E) \in \text{ext}[\mathcal{M}(X_q|x_E)] \right\}. \tag{18}$$

Maximality is in general more informative than interval dominance, i.e., $\mathcal{X}_q^{**} \subseteq \mathcal{X}_q^{*}$. Yet, most of the algorithms for CNs are designed to compute the posterior probabilities as in (16), while the posterior CS is needed by maximality. Notable exceptions are the models considered in classification, for which the computation of the undominated outcomes as in (18) can be performed without explicit evaluation of the posterior CS. Yet, in other cases, the dominance test for any pair of outcomes can be also solved in a CN by simply augmenting the queried node with an auxiliary child and an appropriate quantification of the conditional probabilities.

5.2 Modelling and Updating with Missing Data

The updating problem in (16) refers to a situation where the actual values x_E of the variables X_E are available, while those of the variables in X_M are missing. The latter variables are simply marginalized out. This corresponds to the most popular approach to missing data in the literature and in the statistical practice: the so-called *missing at random* assumption (MAR, [56]), which allows missing data to be neglected, thus turning the incomplete data problem into one of complete data. In particular, MAR implies that the probability of a certain value to be missing does not depend on the value itself, neither on other non-observed values. Yet, MAR is not realistic in many cases, as shown for instance in the following example.

Example 5.1. *Consider the variable* smoker *(S) in Example 1.1. For a given patient, we may want to 'observe' S by simply asking him about that. The outcome of this observation is missing when the patient refuses to answer. MAR corresponds to a situation where the probability that the patient would not answer is independent of whether or not he actually smokes. Yet, it could be realistic to assume that, for instance, the patient is more reluctant to answer when he is a smoker.*

If MAR does not appear tenable, more conservative approaches than simply ignoring missing data are necessary in order to avoid misleading conclusions. De Cooman and Zaffalon have developed an inference rule based on much weaker assumptions than MAR, which deals with near-ignorance about the missingness process [39]. This result has been extended [68] to the case of mixed knowledge about the missingness process: for some variables the process is assumed to be nearly unknown, while it is assumed to be MAR for the others. The resulting updating rule is called *conservative inference rule* (CIR).

To show how CIR-based updating works, we partition the variables in $\mathbf{X}$ in four classes: (i) the queried variable X_q, (ii) the observed variables X_E, (iii) the unobserved MAR variables X_M, and (iv) the variables X_I made missing by a process that we basically ignore. CIR leads to the following CS as our updated beliefs about the queried variable:

19

$$\mathcal{M}(X_q||^{X_I}x_E) \coloneqq \mathrm{CH}\left\{P_j(X_q|x_E, x_I)\right\}_{x_I\in\mathcal{X}_I, j=1,\ldots,v}, \tag{19}$$

where the superscript on the double conditioning bar is used to denote beliefs updated with CIR and to specify the set of missing variables X_I assumed to be non-MAR, and $P_j(X_q|x_E, x_I) = \sum_{x_M} P_j(X_q, x_M|x_E, x_I)$. The insight there is that, as we do not know the actual values of the variables in X_I and we cannot ignore them, we consider all their possible explanations. In particular, when computing lower probabilities, (19) implies:

$$\underline{P}(X_q||^{X_I}x_E) = \min_{x_I\in\mathcal{X}_I} \underline{P}(X_q|x_E, x_I). \tag{20}$$

When coping only with the MAR variables (i.e., if X_I is empty), (20) becomes a standard updating task to be solved by the algorithms in Section 5.3. Although these algorithms cannot be directly applied if X_I is not empty, a procedure to map a CIR task as in (19) into a standard updating task as in (16) for a CN defined over a wider domain has been developed [5].[20] The transformation is particularly simple and consists in the augmentation of the original CN with an auxiliary child for each non-missing-at-random variable, as described by the following example.

Example 5.2 (CIR-based updating by standard algorithms)*.* Consider the CN in Example 4.5. In order to evaluate the probability of a patient having lung cancer (i.e., $C = c$), you perform an X-rays test, and you ask the patient whether he smokes. The X-rays are abnormal (i.e., $R = r$), while, regarding S, the patient refuses to answer (i.e., $S = *$). Following the discussion in Example 5.1, we do not assume MAR for this missing variable. Yet, we do not formulate any particular hypothesis about the reasons preventing the observation of S, so we do CIR-based updating. The problem can be equivalently solved by augmenting the CN with an auxiliary (binary) child O_S of S, such that $\mathcal{M}(O_S|s)$ is a vacuous CS for each $s \in \mathcal{S}$. It is easy to prove that $\underline{P}(c||^S d) = \underline{P}(c|d, o_s)$, where the latter inference can be computed by standard algorithms in the augmented CN. ⧫

5.3 Algorithms for Credal Networks Updating

Despite the hardness of the problem, a number of algorithms for exact updating of CNs have been proposed. Most of these methods generalize existing techniques for Bayesian networks. Regarding Pearl's algorithm for efficient updating on polytree-shaped Bayesian networks [60], a direct extension to CNs is not possible unless all variables are binary. The reason is that a CS over a binary variable has at most two extreme points (see Section 2.1) and it can therefore be identified with an interval. This enables an efficient extension

[19] This updating rule can be applied also to the case of *incomplete* observations, where the outcome of the observation of X_I is missing according to a non-missing-at-random process, but after the observation some of the possible outcomes can be excluded. If $\mathcal{X}'_I \subset \mathcal{X}_I$ is the set of the remaining outcomes, we simply rewrite Equation (19), with $\mathcal{X}'_I$ instead of $\mathcal{X}_I$.

[20] An exhaustive approach to the computation of (20) consisting in the computation of all the lower probabilities on the right-hand side is clearly exponential in the number of variables in X_I.

of Pearl's propagation scheme. The result is an exact algorithm for binary polytree-shaped separately specified CNs, called *2-Updating* (or simply 2U), whose computational complexity is linear in the input size.[21]

Another exception exists if one works with a CN under the concept of *epistemic* irrelevance. In this case, updating can be performed in polynomial time if the topology is a tree [38]. Apart from that, computing lower (and upper) probabilities is an NP-hard problem even in trees. If no constraints are imposed on the topology of the network [30], the problem is not even approximable in polynomial time [34, 58] (in fact this result is shown for a similar problem, but the complexity extends to CN inferences). Hence, for those networks where the inference cannot be processed by an exact method, approximate algorithms come in place and can handle much larger networks [4, 14, 16, 17, 20, 24, 34, 44, 48].

Other approaches to exact inference are also based on generalizations of the most known algorithms for Bayesian networks. For instance, the *variable elimination* technique of Bayesian networks [40] corresponds, in a credal setting, to a *symbolic* variable elimination, where each elimination step defines *multilinear* constraints among the different conditional probabilities where the variable to be eliminated appears. The elimination is said symbolic because numerical calculation are not performed, instead constraints are generated to later be treated by a specialized (non-linear) optimization software. Overall, this corresponds to a mapping between CNs updating as in (16) and *multilinear programming* [13]. Similarly, the *recursive conditioning* technique of Bayesian networks [26] can be used to transform the problem into an integer linear programming problem [31]. Other exact inference algorithms examine potential extreme points of the strong extension according to different strategies in order to produce the required lower/upper values [14, 18], but are very limited in the size of networks that they can handle.

Concerning approximate inference, there are three types: (i) inner approximations, where a (possibly local optimal) solution is returned; (ii) outer approximations, where an outer bound to the objective value is obtained (but no associated feasible solution), and (iii) other methods that cannot guarantee to be inner or outer. Some of these algorithms emphasize enumeration of extreme points, while others resort to non-linear optimization techniques. Outer methods produce intervals that enclose the correct lower and upper probabilities, and are usually based on some relaxation of the original problem. Possible techniques include branch-and-bound methods [28, 31], relaxation of probability values [16], or relaxation of the constraints that define the optimization problem [16]. Inner approximation methods search through the feasible region of the problem using well-known techniques, such as genetic programming [17], simulated annealing [14], hill-climbing [15], or even specialized multilinear optimization methods [19, 20]. Finally, there are methods that cannot guarantee the quality of the result, but usually perform very well in practice. For instance, *loopy propagation* is a popular technique that applies Pearl's propagation to multiply connected Bayesian networks [59]: propagation is iterated until probabilities converge or for a fixed number of iterations. In [47], Ide and Cozman extend these ideas to belief updating on CNs, by developing a loopy variant of 2U that makes

[21] This algorithm has been extended to the case of extensive specifications in [6].

the algorithm usable for multiply connected binary CNs. This idea has been further exploited by the *generalized loopy 2U*, which transforms a generic CN into an equivalent binary CN, which is indeed updated by the loopy version of 2U [4].

5.4 Inference on CNs as a Multilinear Programming Task

In this section we describe by examples two ideas to perform exact inference with CNs: symbolic variable elimination and recursive conditioning. In both cases, the procedure generates a multilinear programming problem, which must be later solved by a specialized software. Multilinear problems are composed of multivariate polynomial constraints and objective where the exponents of optimization variables are either zero or one, that is, each non-linear term is formed by a product of distinct optimization variables.

Example 5.3 (multilinear programming)*.* Consider the task of computing $\underline{P}(s|\neg d)$ in the CN of Example 4.5. In order to perform this calculation, we write a collection of multilinear constraints to be later processed by a multilinear programming solver. The objective function is defined as

$$\min P(s|\neg d) \tag{21}$$

subject to

$$P(s|\neg d) \cdot P(\neg d) = P(s, \neg d), \tag{22}$$

and then two symbolic variable eliminations are used, one with query $\{s, \neg d\}$ and another with query $\{\neg d\}$, to build the constraints that define the probability values appearing in Expression (22). Note that the probability values $P(s|\neg d)$, $P(\neg d)$, and $P(s, \neg d)$ are viewed as optimization variables such that the multilinear programming solver will find the best configuration to minimize the objective function that respects all constraints. Strictly speaking, the minimization is over all the Ps that appear in the multilinear programming problem. Expression (22) ensures that the desired minimum value for $P(s|\neg d)$ is indeed computed as long as the constraints that specify $P(\neg d)$ and $P(s, \neg d)$ represent exactly what is encoded by the CN. The idea is to produce the bounds for $P(s|\neg d)$ without having to explicitly compute the extension of the network.

The symbolic variable elimination that is executed to write the constraints depends on the variable elimination order, just as the bucket elimination in Bayesian networks [40]. In this example, we use the elimination order S, B, C. For the computation of the probability of the event $\{s, \neg d\}$ using that order, the variable elimination produces the following list of computations, which in our case are stored as constraints for the multilinear programming problem (details on variable elimination can be found in [40, 52]):

- Bucket of S: $\forall c', \forall b'$: $\mathbf{P(s, c', b')} = P(s) \cdot P(c'|s) \cdot P(b|s')$.[22] No variable is summed out in this step because S is part of the query. Still, new intermediate values (marked in bold) are defined and will be processed in the next step. In a usual variable elimination, the values $P(s, c', b')$, for every c', b', would be computed and propagated to the next bucket. Here instead optimization variables $P(s, c', b')$ are included in the multilinear programming problem.

[22] Lowercase letters with a prime are used to indicate a generic state of a binary variable.

- Bucket of B: $\forall c'$: $\mathbf{P}(\mathbf{s}, \mathbf{c}', \neg\mathbf{d}) = \sum_b \mathbf{P}(\mathbf{s}, \mathbf{c}', \mathbf{b}) \cdot P(\neg d|c', b)$. In this bucket of B is summed out (eliminated) in the probabilistic interpretation. New intermediate values $P(s, c', \neg d)$ (for every c') appear and will be dealt in the next bucket (again, in a usual variable elimination, they would be the propagated values).

- Bucket of C: $\mathbf{P}(\mathbf{s}, \neg\mathbf{d}) = \sum_{c'} \mathbf{P}(\mathbf{s}, \mathbf{c}', \neg\mathbf{d})$. By this equation C is summed out, obtaining the desired result. In the multilinear interpretation, the optimization variables $P(s, c', \neg d)$ are employed to form a constraint that defines $P(s, \neg d)$.

In bold we highlight the intermediate probability values that are not part of the CN specification, so they are solely tied by the equations just presented to the probability values that are part of the input (those not in bold). Overall, these equations define the value of $P(s, \neg d)$ in terms of the input values of the problem.

The very same idea is employed in the symbolic computation of the probability of $\neg d$:

- Bucket of S: $\forall c', \forall b'$: $\mathbf{P}(\mathbf{c}', \mathbf{b}') = \sum_{s'} P(s') \cdot P(c'|s') \cdot P(b'|s')$. Now S is not part of the query, so it is summed out. Four constraints (one for each joint configuration c', b') define the values $P(c', b')$ in terms of the probability values involving S.

- Bucket of B: $\forall c'$: $\mathbf{P}(\mathbf{c}', \neg\mathbf{d}) = \sum_{b'} \mathbf{P}(\mathbf{c}', \mathbf{b}') \cdot P(\neg d|c', b')$. Here B is summed out, and the (symbolic) result is $P(c', \neg d)$, for each c'.

- Bucket of C: $\mathbf{P}(\neg\mathbf{d}) = \sum_{c'} \mathbf{P}(\mathbf{c}', \neg\mathbf{d})$. This final step produces the glue between values $P(c', \neg d)$ and $P(\neg d)$ by summing out C.

Finally, the constraints generated by the symbolic variable elimination are put together with those forcing the probability mass functions to lie inside their local CSs, which are simply those specified in Example 4.5. All these constraints are doing is to formalise the dependence relations between the variables in the network, as well as the local CS specifications. Clearly, a different elimination ordering would produce a different multilinear program leading to the same solution. The chosen elimination order has generated terms with up to three factors, that is, polynomials of order three. ⧫

We illustrate another idea to generate constraints based on multilinear programming, which uses the idea of conditioning. Instead of a variable elimination, we keep an active set of conditioning variables that cut the network graph, in the same manner as done by the recursive conditioning method for Bayesian networks.

Example 5.4 (conditioning). Let us evaluate $P(\neg d)$ and write the constraints that define it, we have:

- Cut-set $\{S\}$: $P(\neg d) = \sum_{s'} P(s') \cdot \mathbf{P}(\neg\mathbf{d}|\mathbf{s}')$. In this step, the probability of D is conditioned on S by one constraint. New intermediate probability values arise (in bold) that are not part of the network specification. Next step takes $P(\neg d|s')$ to be processed and defined through other constraints.

- Cut-set $\{S, C\}$: $\forall s'$: $P(\neg d|s') = \sum_{c'} P(c'|s') \cdot \mathbf{P}(\neg\mathbf{d}|\mathbf{s}', \mathbf{c}')$. The probability of $D|S$ is conditioned on C by two constraints (one for each value of S). Again, new intermediate values appear in bold, which are going to be treated in the next step.

- Cut-set $\{C, B\}$: $\forall s', \forall c' : \; P(\neg d|s', c') = \sum_{b'} P(b'|s') \cdot P(\neg d|c', b')$. Here $D|S, C$ is further conditioned on B by using four constraints (one for each value of S and C), which leaves only C and B in the cut-set (D is independent of S given C, B). The probability values that appear are all part of the network specification, and thus we may stop to create constraints. $P(\neg d)$ is completely written as a set of constraints over the input values.

As in the previous example, the cut-set constraints are later put together with the local constraints of the CSs, as well as the constraints to specify $P(s, \neg d)$ (in this example, this last term is easily defined by a single constraint: $P(s, \neg d) = P(s) \cdot P(\neg d|s)$, because the latter element had already appeared in the cut-set constraints for the cut $\{S, C\}$ and thus is already well-defined by previous constraints). ⧫

The construction of the multilinear programming problem just described takes time proportional to an inference in the corresponding precise Bayesian network. The great difference is that, after running such transformation, we still have to optimize the multilinear problem, while in the precise case the result would already be available. Hence, to complete the example, we have run an optimizer over the problems constructed here to obtain $\underline{P}(s|\neg d) = 0.1283$. Replacing the minimization by a maximization, we get $\overline{P}(s|\neg d) = 0.4936$. Performing the same transformation for d instead of $\neg d$, we obtain $P(s|d) \in [0.2559, 0.7074]$, that is, the probability of smoking given dyspnoea is between one fourth and seventy percent, while smoking given *not* dyspnoea is between twelve and forty-nine percent. The source code of the optimization problems that are used here are available online in the address `http://ipg.idsia.ch/`. The reader is invited to try them out.

6 Further Reading

We conclude this chapter by surveying a number of challenges, open problems, and alternative models to those we have presented here.

We start with a discussion on probabilistic graphical models with imprecision other than credal networks with strong independence. As noted in Section 5.3, the literature has recently started exploring an alternative definition of credal network where strong independence is replaced by the weaker concept of *epistemic irrelevance* [38] (some earlier work in this sense was also done in [33]). This change in the notion of independence used by a credal network affects the results of the inferences [38, Section 8] even if the probabilistic information with which one starts is the same in both cases (in particular the inferences made under strong independence will be never less precise, and typically more precise, than those obtained under irrelevance). This means that it is very important to choose the appropriate notion of independence for the domain under consideration.

Yet, deciding which one is the ‘right’ concept for a particular problem is not always clear. A justification for using strong independence may rely on a sensitivity analysis interpretation of imprecise probabilities: one assumes that some ‘ideal’ precise probability satisfying stochastic independence exists, and that, due to the lack of time or other resources, can only be partially specified or assessed, thus giving rise to sets of models that satisfy stochastic independence. Although this seems to be a useful interpretation in a number of problems, it is not always applicable. For instance, it is questionable that expert knowledge should comply with the sensitivity analysis interpretation.

Epistemic irrelevance has naturally a broader scope, as it only requires that some variables are judged not to influence other variables in a model. For this reason, research on epistemic irrelevance is definitely a very important topic in the area of credal networks. On the other hand, at the moment we know relatively little about how practical is using epistemic irrelevance. The paper mentioned above [38] sheds a positive light on this, as it shows that tree-shaped credal networks based on irrelevance can be updated very easily. This, for instance, is not the case of trees under strong independence [58]. But that paper also shows that irrelevance can frequently give rise to dilation [62] in a way that may not always be desirable. This might be avoided using the stronger, symmetrized, version of irrelevance called epistemic independence. But the hope to obtain efficient algorithms under this stronger notion is much less that under irrelevance. Also, epistemic irrelevance and independence have been shown to make some of the graphoid axioms fail [21], which is an indication that the situation on the front of efficient algorithms could become complicated on some occasions.

In this sense, the situation of credal networks under strong independence is obviously much more consolidated, as research on this topic has been, and still is, intense, and has been going on for longer.[23] Moreover, the mathematical properties of strong independence make it particularly simple to represent a credal network as a collection of Bayesian networks, and this makes quite natural to (try to) extend algorithms originally developed for Bayesian networks into the credal setting. This makes it easier, at the present time, to address applications using credal networks under strong independence.

In summary, we believe that it is too early to make any strong claims on the relative benefits of the two concepts, and moreover we see the introduction of models based on epistemic irrelevance as an exciting and important new avenue for research on credal networks.

Another challenge concerns the development of credal networks with continuous variables. Benavoli et. al [10] have proposed an imprecise hidden Markov model with continuous variables using Gaussian distributions, which produces a reliable Kalman filter algorithm. This can be regarded as a first example of credal network with continuous variables over a tree topology. Similarly, the framework for the fusion of imprecise probabilistic knowledge proposed in [9] corresponds to a credal network with continuous variables over the naive topology. These works use coherent lower previsions for the general inference algorithms, and also provide a specialized version for linear-vacuous mixtures. The use of continuous variables within credal networks is an interesting topic and deserves future attention.

Decision trees have been also explored [46, 50, 51, 64, 65]. Usually in an imprecise decision tree, the decision nodes and utilities are treated in the same way as in their precise counterpart, while chance nodes are filled with imprecise probabilistic assessments.

[23] The first formalisation of the notion of credal network was based on strong independence [18].

The most common task is to find the expected utility of a decision or find the decisions (or *strategies*) that maximize the expected utility. However, the imprecision leads to imprecise expected utilities, and distinct decision criteria can be used to select the best strategy (or set of strategies). By some (reasonably simple) modifications of the tree structures, it is possible to obtain a credal network (which is not necessarily a tree) whose inference is equivalent to that of the decision tree. In a different approach, where the focus is on classification, imprecise decision trees have been used to build more reliable *classifiers* [1].

Qualitative and semi-qualitative networks, which are Bayesian networks extended with qualitative assessments about the probability of events, are also a type of credal network. The (semi-)qualitative networks share most of the characteristics of a credal network: the set of random variables, the directed acyclic graph with a Markov condition, and the local specification of conditional probability mass functions in accordance with the graph. However, these networks admit only some types of constraints to specify the local credal sets. For example, qualitative *influences* define that the probability of a state s of a variable is greater given one parent instantiation than given another, which indicates that a given observed parent state implies a greater chance of seeing s. Other qualitative relations are *additive* and *multiplicative* synergies. The latter are non-linear constraints and can be put within the framework of credal sets by extending some assumptions (for example, we cannot work only with finitely many extreme points). Qualitative networks have only qualitative constraints, while semi-qualitative networks also allow mass functions to be numerically defined. Some inferences in qualitative networks can be processed by fast specialized algorithms, while inferences in semi-qualitative networks (mixing qualitative and quantitative probabilistic assessments) are as hard as inferences in general credal networks [29, 32].

Markov decision processes have received considerable attention under the theory of imprecise probability [49, 63]. In fact, the framework of Markov decision process has been evolved to deal with deterministic, non-deterministic and probabilistic planning [45]. This has happened in parallel with the development of imprecise probability, and recently it has been shown that Markov decision processes with imprecise probability can have precise and imprecise probabilistic transitions, as well as set-valued transitions, thus encompassing all those planning paradigms [43]. Algorithms to efficient deal with Markov decision processes with imprecise probabilities have been developed [41, 49, 51, 63, 66].

Other well-known problems in precise models have been translated to inferences in credal networks in order to exploit the ideas of the latter to solve the former. For instance, the problem of strategy selection in influence diagrams and in decision networks was mapped to a query in credal networks [36]. Most probable explanations and maximum a posteriori problems of Bayesian networks can be also easily translated into credal networks inferences [30]. Inferences in probabilistic logic, when augmented by stochastic irrelevance/independence concepts, naturally become credal network inferences [22, 23, 35]. Other extensions are possible, but still to be done. For example, dynamic credal networks have been mentioned in the past [42], but are not completely formalised and widely used. Still, hidden Markov models are a type dynamic Bayesian networks, so the same relation exists in the credal setting. Besides imprecise hidden Markov models, dynamic credal networks have appeared to model the probabilistic relations of decision trees and Markov decision processes. Moreover, undirected probabilistic graphical models (such as Markov random fields) can clearly be extended to imprecise probabilities. Markov random fields are described by an undirected graph where local functions are defined over the variables that belong to a same clique. These functions are not constrained to be probability distribution, as the whole network is later normalized by the so called partition function. Hence, the local functions can be made imprecise in order to build an imprecise Markov random field, on which inferences would be more reliable.

Overall, a number of probabilistic graphical models with imprecision other than credal networks has been proposed in the literature. We devoted most of this chapter to credal networks because their theoretical development is already quite mature, thus making it possible to show the expressive power (as well as the computational challenges) of approaches based on imprecise probabilities. Furthermore, credal networks have been already applied in a number of real-world problems for the implementation of knowledge-based expert systems (see [2, 3, 37] for some examples, and [61] for a tutorial on implementing these applications). Applications to classification will be considered in the next chapter.

References

[1] Joaquín Abellán and Andrés Masegosa. Combining decision trees based on imprecise probabilities and uncertainty measures. In Khaled Mellouli, editor, *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, volume 4724 of *Lecture Notes in Computer Science*, pages 512–523. Springer Berlin / Heidelberg, 2007.

[2] A. Antonucci, R. Brühlmann, A. Piatti, and M. Zaffalon. Credal networks for military identification problems. *International Journal of Approximate Reasoning*, 50(2):666–679, 2009.

[3] A. Antonucci, A. Salvetti, and M. Zaffalon. Credal networks for hazard assessment of debris flows. In J. Kropp and J. Scheffran, editors, *Advanced Methods for Decision Making and Risk Management in Sustainability Science.* Nova Science Publishers, New York, 2007.

[4] A. Antonucci, S. Yi, C.P. de Campos, and M. Zaffalon. Generalized loopy 2U: a new algorithm for approximate inference in credal networks. *International Journal of Approximate Reasoning*, 51(5):474–484, 2010.

[5] A. Antonucci and M. Zaffalon. Equivalence between Bayesian and credal nets on an updating problem. In J. Lawry, E. Miranda, A. Bugarin, S. Li, M. A. Gil, P. Grzegorzewski, and O. Hryniewicz, editors, *Proceedings of third international conference on Soft Methods in Probability and Statistics (SMPS-2006)*, pages 223–230. Springer, 2006.

[6] A. Antonucci and M. Zaffalon. Decision-theoretic specification of credal networks: A unified language for uncertain modeling with sets of bayesian networks. *International Journal of Approximate Reasoning*, 49(2):345–361, 2008.

[7] Thomas Augustin, Frank P. A. Coolen, Serafin Moral, and Matthias C. M. Troffaes, editors. *ISIPTA '09: Proceedings of the Sixth International Symposium on Imprecise Probabilities: Theories and Applications*, Durham, United Kingdom, 2009. SIPTA.

[8] D. Avis and K. Fukuda. A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra. *Discrete and Computational Geometry*, 8:295–313, 1992.

[9] A. Benavoli and A. Antonucci. Aggregating Imprecise Probabilistic Knowledge: application to Zadeh's paradox and sensor networks. *International Journal of Approximate Reasoning*, accepted 2010.

[10] A. Benavoli, M. Zaffalon, and E. Miranda. Reliable hidden Markov model filtering through coherent lower previsions. In *Proc. 12th Int. Conf. Information Fusion*, pages 1743–1750, Seattle (USA), 2009.

[11] James O. Berger. The robust Bayesian viewpoint. In J. B. Kadane, editor, *Robustness of Bayesian Analyses*, pages 63–144. Elsevier Science, Amsterdam, 1984.

[12] George Boole. *An investigation of the laws of thought on which are founded the mathematical theories of logic and probabilities.* Walton and Maberly, London, 1854.

[13] L. Campos, J. Huete, and S. Moral. Probability intervals: a tool for uncertain reasoning. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 2(2):167–196, 1994.

[14] A. Cano, J. Cano, and S. Moral. Convex sets of probabilities propagation by simulated annealing on a tree of cliques. In *Proceedings of Fifth International Conference on Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU '94)*, pages 4–8, 1994.

[15] A. Cano, M. Gómez, and S. Moral. Application of a hill-climbing algorithm to exact and approximate inference in credal networks. In F. G. Cozman, B. Nau, and T. Seidenfeld, editors, *ISIPTA '05: Proceedings of the Fourth International Symposium on Imprecise Probabilities and Their Applications*, pages 88–97, Pittsburgh, USA, 2005.

[16] A. Cano and S. Moral. Using probability trees to compute marginals with imprecise probabilities. *International Journal of Approximate Reasoning*, 29(1):1–46, 2002.

[17] S. Cano, A.and Moral. A genetic algorithm to approximate convex sets of probabilities. In *Proceeding of the Six International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU-96)*, volume II, pages 847–852, 2009.

[18] F. G. Cozman. Credal networks. *Artificial Intelligence*, 120:199–233, 2000.

[19] F. G. Cozman and C. P. de Campos. Local computation in credal networks. In *Workshop on Local Computation for Logics and Uncertainty*, pages 5–11, Valencia, 2004. IOS Press.

[20] F. G. Cozman, C. P. de Campos, J. S. Ide, and J. C. F. da Rocha. Propositional and relational Bayesian networks associated with imprecise and qualitative probabilistic assessments. In *Conference on Uncertainty in Artificial Intelligence*, pages 104–111, Banff, 2004. AUAI Press.

[21] Fabio G. Cozman and Peter Walley. Graphoid properties of epistemic irrelevance and independence. *Annals of Mathematics and Artificial Intelligence*, 45:173–195, October 2005.

[22] F.G. Cozman, C.P. de Campos, and J.C.F. da Rocha. Probabilistic logic with independence. *International Journal of Approximate Reasoning*, 49(1):3–17, 2008.

[23] F.G. Cozman and R.B. Polastro. Complexity analysis and variational inference for interpretation-based probabilistic description logics. In *Proceeding of the 25th Conference on Uncertainty in Artificial Intelligence*, pages 120–133, 2009.

[24] J. C. da Rocha, F. G. Cozman, and C. P. de Campos. Inference in polytrees with sets of probabilities. In *Conference on Uncertainty in Artificial Intelligence*, pages 217–224, Acapulco, 2003.

[25] G. B. Dantzig. *Linear programming and extensions.* Rand Corporation Research Study. Princeton University Press, Princeton, NJ, 1963.

[26] A. Darwiche. Recursive conditioning. *Artificial Intelligence*, 126(1-2):5–41, 2001.

[27] A. P. Dawid. Conditional independence in statistical theory. *Journal of the Royal Statistical Society. Series B (Methodological)*, 41(1):1–31, 1979.

[28] C. P. de Campos and F. G. Cozman. Inference in credal networks using multilinear programming. In *Proceedings of the Second Starting AI Researcher Symposium*, pages 50–61, Amsterdam, 2004. IOS Press.

[29] C. P. de Campos and F. G. Cozman. Belief updating and learning in semi-qualitative probabilistic networks. In *Conference on Uncertainty in Artificial Intelligence*, pages 153–160, 2005.

[30] C. P. de Campos and F. G. Cozman. The inferential complexity of Bayesian and credal networks. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1313–1318, Edinburgh, 2005.

[31] C. P. de Campos and F. G. Cozman. Inference in credal networks through integer programming. In *Proceedings of the Fifth International Symposium on Imprecise Probability: Theories andApplications*, Prague, 2007. Action M Agency.

[32] C. P. de Campos, L. Zhang, Y. Tong, and Q. Ji. Semi-qualitative probabilistic networks in computer vision problems. *Journal of Statistical Theory and Practice*, 3(1):197–210, 2009.

[33] Cassio Polpo de Campos and Fabio Gagliardi Cozman. Computing lower and upper expectations under epistemic independence. *International Journal of Approximate Reasoning*, 44(3):244 – 260, 2007. Reasoning with Imprecise Probabilities.

[34] C.P. de Campos. New results for the map problem in bayesian networks. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2100–2106. AAAI Press, 2011.

[35] C.P. de Campos, F.G. Cozman, and Luna J.E.O. Assembling a consistent set of sentences in relational probabilistic logic with stochastic independence. *Journal of Applied Logic*, 7(2):137–154, 2009.

[36] C.P. de Campos and Q. Ji. Strategy selection in influence diagrams using imprecise probabilities. In *Proceedings of the 24th Conference in Uncertainty in Artificial Intelligence, July 9-12, 2008, Helsinki, Finland*, pages 121–128, 2008.

[37] C.P. de Campos, L. Zhang, Y. Tong, and Q. Ji. Semi-qualitative probabilistic networks in computer vision problems. In P. Coolen-Schrijner, F. Coolen, M.C.M. Troffaes, and T. Augustin, editors, *Imprecision in statistical theory and practice.*, pages 207–220. Grace Scientific Publishing LLC, Greensboro, North-Carolina, USA, 2009.

[38] G. de Cooman, F. Hermans, A. Antonucci, and M. Zaffalon. Epistemic irrelevance in credal networks: the case of imprecise Markov trees. *International Journal of Approximate Reasoning*, accepted for publication.

[39] G. de Cooman and M. Zaffalon. Updating beliefs with incomplete observations. *Artificial Intelligence*, 159:75–125, 2004.

[40] R. Dechter. Bucket elimination: A unifying framework for probabilistic inference. In Eric Horvitz and Finn Jensen, editors, *Conference on Uncertainty in Artificial Intelligence*, pages 211–219, San Francisco, 1996. Morgan Kaufmann Publishers.

[41] K.V. Delgado, L.N. de Barros, F.G. Cozman, and Shirota R. Representing and solving factored markov decision processes with imprecise probabilities. In Augustin et al. [7], pages 169–178.

[42] K.V. Delgado, S. Sanner, L.N. de Barros, and F.G. Cozman. Efficient solutions to factored mdps with imprecise transition probabilities. In *Proceedings of the Nineteenth International Conference on Automated Planning and Scheduling (ICAPS-09)*, pages 98–105, 2009.

[43] Leliane N. de Barros Felipe W. Trevizan, Fabio G. Cozman. Mixed probabilistic and nondeterministic factored planning through markov decision processes with set-valued transitions. In *Workshop on A Reality Check for Planning and Scheduling Under Uncertainty at the Eighteenth International Conference on Automated Planning and Scheduling (ICAPS)*, 2008.

[44] J.C. Ferreira da Rocha and F.G. Cozman. Inference in credal networks: branch-and-bound methods and the a/r+ algorithm. *International Journal of Approximate Reasoning*, 39(2-3):279–296, 2005.

[45] Robert Givan, Sonia Leach, and Thomas Dean. Bounded parameter markov decision processes. In Sam Steel and Rachid Alami, editors, *Recent Advances in AI Planning*, volume 1348 of *Lecture Notes in Computer Science*, pages 234–246. Springer Berlin / Heidelberg, 1997.

[46] Nathan Huntley and Matthias Troffaes. An efficient normal form solution to decision trees with lower previsions. In Didier Dubois, M. Lubiano, Henri Prade, MarÃŋa Gil, Przemyslaw Grzegorzewski, and Olgierd Hryniewicz, editors, *Soft Methods for Handling Variability and Imprecision*, volume 48 of *Advances in Soft Computing*, pages 419–426. Springer Berlin / Heidelberg, 2008.

[47] J. S. Ide and F. G. Cozman. IPE and L2U: Approximate algorithms for credal networks. In *Proceedings of the Second Starting AI Researcher Symposium*, pages 118–127, Amsterdam, 2004. IOS Press.

[48] J.S. Ide and F.G. Cozman. Approximate algorithms for credal networks with binary variables. *Int. J. Approx. Reasoning*, 48(1):275–296, 2008.

[49] Hideaki Itoh and Kiyohiko Nakamura. Partially observable markov decision processes with imprecise parameters. *Artif. Intell.*, 171(8-9):453–490, 2007.

[50] Gildas Jeantet and Olivier Spanjaard. Optimizing the hurwicz criterion in decision trees with imprecise probabilities. In *ADT '09: Proceedings of the 1st International Conference on Algorithmic Decision Theory*, pages 340–352, Berlin, Heidelberg, 2009. Springer-Verlag.

[51] D. Kikuti, F. G. Cozman, and C. P. de Campos. Partially ordered preferences in decision trees: Computing strategies with imprecision in probabilities. In *IJCAI Workshop about Advances on Preference Handling*, pages 1313–1318, 2005.

[52] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques.* MIT Press, 2009.

[53] Vladimir P. Kuznetsov. *Interval Statistical Models.* Radio i Svyaz Publ., Moscow, 1991. In Russian.

[54] S. L. Lauritzen and D. 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.

[55] Isaac Levi. *The Enterprise of Knowledge. An Essay on Knowledge, Credal Probability, and Chance.* MIT Press, Cambridge, 1980.

[56] R. J. A. Little and D. B. Rubin. *Statistical Analysis with Missing Data.* Wiley, New York, 1987.

[57] M. L. Littman, J. Goldsmith, and M. Mundhenk. The computational complexity of probabilistic planning. *Journal of Artificial Intelligence Research*, 9:1–36, 1998.

[58] Denis D. Maua, Cassio P. de Campos, Alessio Benavoli, and Alessandro Antonucci. On the complexity of strong and epistemic credal networks. In *29th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 391–400. AUAI Press, 2013.

[59] K. Murphy, Y. Weiss, and M. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *Conference on Uncertainty in Artificial Intelligence*, pages 467–475, San Francisco, 1999. Morgan Kaufmann.

[60] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, San Mateo, California, 1988.

[61] A. Piatti, A. Antonucci, and M. Zaffalon. Building knowledge-based systems by credal networks: a tutorial. In A. R. Baswell, editor, *Advances in Mathematics Research.* Nova Science Publishers, New York, 2010.

[62] T. Seidenfeld and L. Wasserman. Dilation for sets of probabilities. *The Annals of Statistics*, 21:1139–54, 1993.

[63] R. Shirota, F. Cozman, F. W. Trevizan, and C. P. de Campos. Multilinear and integer programming for markov decision processes with imprecise probabilities. In *5th International Symposium on Imprecise Probability: Theories and Applications*, Prague, 2007.

[64] R. Shirota, D. Kikuti, and F.G. Cozman. Solving decision trees with imprecise probabilities through linear programming. In Augustin et al. [7].

[65] Matthias C. M. Troffaes, Nathan Huntley, and Ricardo Shirota Filho. Sequential decision processes under act-state independence with arbitrary choice functions. In Eyke Hüllermeier, Rudolf Kruse, and Frank Hoffmann, editors, *Information Processing and Management of Uncertainty in Knowledge-Based Systems. Theory and Methods*, volume 80 of *Communications in Computer and Information Science*, pages 98–107. Springer Berlin Heidelberg, 2010.

[66] M.C.M. Troffaes. Learning and optimal control of imprecise markov decision processes by dynamic programming using the imprecise dirichlet model. In M. Lopéz-Díaz, M.A. Gil, P. Grzegorzewski, O. Hyrniewicz, and Lawry, editors, *Soft Methodology and Random Information Systems*, pages 141–148. Springer Berlin / Heidelberg, 2004.

[67] Peter Walley. *Statistical Reasoning with Imprecise Probabilities*, volume 42 of *Monographs on Statistics and Applied Probability.* Chapman and Hall, London, 1991.

[68] M. Zaffalon and E. Miranda. Conservative Inference Rule for Uncertain Reasoning under Incompleteness. *Journal of Artificial Intelligence Research*, 34:757–821, 2009.