Imprecise Probabilistic Graphical Models: Equivalent Representations, Inference Algorithms and Applications

Doctoral Dissertation submitted to the
Faculty of Informatics of the University of Lugano
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

presented by
Alessandro Antonucci

under the supervision of
Marco Zaffalon

April 2008

Dissertation Committee

Gert de Cooman	Ghent University, Belgium
Fabio Crestani	University of Lugano, Switzerland
Luca Maria Gambardella	IDSIA, Switzerland
Serafín Moral	University of Granada, Spain
Marco Zaffalon	IDSIA, Switzerland

Dissertation accepted on 16 April 2008

Supervisor	PhD program director
Marco Zaffalon	**Fabio Crestani**

I certify that except where due acknowledgement has been given, the work presented in this thesis is that of the author alone; the work has not been submitted previously, in whole or in part, to qualify for any other academic award; and the content of the thesis is the result of work which has been carried out since the official commencement date of the approved research program.

Alessandro Antonucci
Lugano, 16 April 2008

In memory of my father

Abstract

Credal networks are probabilistic graphical models that extend Bayesian nets to deal with imprecision in probability, and can actually be regarded as sets of Bayesian nets. Credal nets appear to be powerful means to represent and deal with many important and challenging problems in uncertain reasoning. The counterpart of having more freedom in the modeling phase is an increased inferential complexity of inferences, e.g., the so-called *belief updating* becomes a hard task even on relatively simple topologies.

In this thesis, I start my investigation on credal networks by considering equivalent representations of those models. More specifically, I first deliver a new graphical language, which is called *decision-theoretic* being inspired by the formalism of decision graphs, for a unified representation of credal networks of any kind. I also present another representation, which is called *binarization*, being in fact a reformulation of a credal network solely based on binary variables. Remarkably, I prove that if a credal net is first reformulated by its decision-theoretic representation and then by the corresponding binarization, the resulting representation is completely equivalent. An equivalence relation between Bayesian and credal nets, when the reason for the missingness of some of the variables in the Bayesian nets is unknown, is also provided.

The developed equivalent representations are applied to inference problems. First, I show that, by a decision-theoretic formulation, the algorithms that have been already designed for credal networks, which are mostly referred to a specific class of models, called *separately specified* nets, can be generalized to credal networks of any kind. Similar formalisms are also employed to solve inference and classification problems with missing observations. I also present a state-of-the-art updating algorithm which is based on the equivalent binary representation. This algorithm, called GL2U, offers an efficient procedure for approximate updating of general credal nets. The quality of the overall approximation is investigated by promising numerical experiments. As a further theoretical investigation, I consider a classification problem for Bayesian networks for which a hardness proof together with a fast algorithm for a subclass of models is provided.

Finally, two real-world applications of credal networks are presented. First, I consider a military identification problem, consisting in the detection of the goal of an intruder entering a no-fly area. The problem, together with the necessary fusion of the information gathered by the sensors is mapped by our techniques into a credal network updating task. The solution is then obtained by the GL2U algorithm. The second application is an environmental model for hazard assessment of debris flows by credal networks. A credal network evaluates the level of risk, corresponding to the observed values of the triggering factors, for this specific natural hazard. For some factors, whose observations are more difficult, the corresponding soft evidential information is embedded by our formalism into the structure of the network. This model is employed for extensive numerical analysis on the Swiss territory.

Acknowledgements

First I want to thank those persons that had a direct influence on the contents of this thesis. These include first and foremost my supervisor, Marco Zaffalon; his support in terms of academic discussions, "pedagogical" directions and human nearness is something I will treasure for the rest of my life. Then, the people whom I have collaborated with on various papers. These are Fabio Gagliardi Cozman, who had a very strong influence on my view of probabilistic graphical models, Alberto Piatti, Andrea Salvetti and Ralph Brühlmann, for helping me in widening my interests from purely theoretical analysis towards applications, and Jaime Shinsuke Ide and Cassio Polpo de Campos, partners of exciting researches.

Furthermore, I would like to thank all the people at IDSIA. It was a real privilege for me to be part of such a great group of persons. I want to mention Giorgio Corani and Sun Yi for the discussions about imprecise probabilities, Leonora Bianchi for the support during my first weeks at IDSIA, and Luca Maria Gambardella for his help.

I am also grateful to the people from the SIPTA community, the symposia and summer schools they organize were extremely important to me in terms of ideas and enthusiasm. Among the members, I am especially grateful to Gert de Cooman and Enrique Miranda, because their work helped me in perceiving the beauty of the theory of imprecise probabilities. Finally, I thank my wife Vita for being with me all these years, especially during the difficult times.

I also acknowledge the Swiss National Science Foundation (SNF), IDSIA and SUPSI-DTI, and Armasuisse for their financial support. More specifically the research presented in this thesis has been supported by the following grants:

- SNF grant 2100-067961.02. *Investigations on the theory and practice of credal classification*.
- SNF grant 200020-109295/1. *Uncertain reasoning under incompleteness*.
- Armasuisse grant R-3210/044-01. *CREDO: Identifizierung von unbekannten Objekten mittels kredaler Klassifizierung*.
- Armasuisse grant 044-04. *CREDO Phase II: Sensornetzwerke für Sicherungsoperationen*.

Contents

Figures

Tables

Chapter 1

Introduction

This thesis represents a general investigation into the field of credal networks, that ranges from purely theoretical analysis, towards applications to inference and classification problems, and hence to real-world applications. Credal networks are probabilistic graphical models that extend Bayesian nets to deal with imprecision, and can actually be regarded as sets of Bayesian nets. Credal nets appear to be powerful means to represent and deal with many important and challenging problems in uncertain reasoning. The counterpart of having more freedom in the modeling phase is an increased complexity of inferences.

1.1 Main Scientific Contributions

The main results presented in this thesis can be summarized as follows:

- A new graphical language, which is called decision-theoretic being inspired by the formalism of decision graphs, for a unified representation of credal networks of any kind.

- Another representation, called binarization, which is in fact an equivalent reformulation of general credal networks solely based on binary variables.

- A state-of-the-art updating algorithm which is based on our equivalent binary representation. This algorithm, called GL2U, offers an efficient procedure for approximate updating of general credal nets.

- A fast algorithm for classification on both Bayesian and credal networks when some of the observed variables are missing according to a mechanism that is ignored.

- Two real-world applications of our formalisms and algorithms for credal networks, that allow for addressing a military identification problem and an environmental risk analysis task.

1.2 Organization of the Thesis

Let us quickly outline the structure of the thesis. After this first introductory chapter, we have a background chapter where general definitions and standard results about Bayesian and credal networks are reported. In Chapter 3 we obtain an equivalence relation between Bayesian and credal networks with respect to two different updating problems. This result together with other important problems in uncertain reasoning, which are reported in the same chapter, suggests the need for a unified formalism for general credal networks, which is provided by the graphical language defined in Chapter 4. Chapter 5 describes a new updating algorithm based on this language and the related numerical tests. Chapter 6 moves from updating to classification problems and provides some complexity results and fast algorithms for Bayesian and credal networks with incomplete observation of the variables. Finally, Chapters 7 and 8 describe two real-world applications of credal networks, referred respectively to a military identification problem and to an environmental hazard assessment problem.

1.3 List of Papers

This thesis is based on theoretical, numerical and applied research which has been written up in twelve scientific papers, which have passed the peer-review process and been accepted for publication in international journals, books chapters, and proceedings of various international conferences, symposia, and workshops with high academic standards, and two recent papers that are still under review. This section lists the papers on which the thesis is based, along with in what section or chapter the results in each paper is discussed. Note that most of the papers share at least some theory, related work, and methods, and these are discussed in Chapter 2. The papers are:

- International Journals

  - Antonucci, A., Brühlmann, R., Piatti, A., Zaffalon., M. (submitted). Credal networks for military identification problems. *International Journal of Approximate Reasoning*. Briefly discussed in Chapter 7.

  - Antonucci, A., Zaffalon., M. (accepted for publication). Decision-theoretic specification of credal networks: a unified language for uncertain modeling with sets of Bayesian networks. *International Journal of Approximate Reasoning*. Briefly discussed in Chapters 3 and 4.
  - Antonucci, A., Zaffalon., M. (2007). Fast algorithms for robust classification with Bayesian nets. *International Journal of Approximate Reasoning*. **44**(3), 200–223. Briefly discussed in Chapter 6.

- International Conferences and Workshops

  - Antonucci, A., Zaffalon, M., Sun, Y., de Campos, C. P. (2008). Generalized loopy 2U: a new algorithm for approximate inference in credal networks. In Jaeger, M., Nielsen, T. D. (Eds), *PGM'08: Proceedings of the Fourth European Workshop on Probabilistic Graphical Models*. Hirtshals (Denmark), pp. 17–24. Briefly discussed in Chapter 5.
  - Antonucci, A., Brühlmann, R., Piatti, A., Zaffalon, M. (2007). Credal networks for military identification problems. In de Cooman, G., Vejnarová, J., Zaffalon, M. (Eds), *Proceedings of the Fifth International Symposium on Imprecise Probability: Theories and Applications* (ISIPTA '07), pp. 1–10. Prague (Czech Republic). Action M Agency. Briefly discussed in Chapter 7.
  - Antonucci, A., Zaffalon., M. (2006). Locally specified credal networks. In Studený, M., Vomlel, J. (Eds), *Proceedings of the third European Workshop on Probabilistic Graphical Models (PGM-2006)*, pp. 25–34. Prague, (Czech Republic). Action M Agency. Briefly discussed in Chapters 3 and 4.
  - Antonucci, A., Zaffalon, M. (2006). Equivalence between Bayesian and credal nets on an updating problem. In Lawry, J., Miranda, E., Bugarin, A., Li, S., Gil, M.A., Grzegorzewski, P., Hryniewicz, O. (Eds), *Soft Methods for Integrated Uncertainty Modeling* (Proceedings of the third international conference on Soft Methods in Probability and Statistics: SMPS 2006), pp. 223–230. Springer. Briefly discussed in Section 3.1.
  - Antonucci, A., Zaffalon, M., Ide, J. S., Cozman, F. G. (2006). Binarization algorithms for approximate updating in credal nets. In Penserini, L., Peppas, P., Perini, A. (Eds), *Proceedings of the third European Starting AI Researcher Symposium (STAIRS-2006)*, pp. 120–131. Amsterdam, Netherlands. IOS Press. Briefly discussed in Section 5.1.

  - Antonucci, A., Zaffalon, M. (2005). Fast algorithms for robust classification with Bayesian nets. In Cozman, F. G., Nau, R., Seidenfeld, T. (Eds), *Proceedings of the fourth International Symposium on Imprecise Probabilities and Their Applications (ISIPTA '05)*, pp. 11–20. SIPTA. Briefly discussed in Chapter 6.
  - Antonucci, A., Salvetti, A., Zaffalon, M. (2004). Assessing debris flow hazard by credal nets. In Lopez-Diaz, M., Gil, M. A., Grzegorzewski, P., Hryniewicz, O., Lawry, J. (Eds), *Proceedings of the Second International Conference on Soft Methods in Probability and Statistics (SMPS-2004) - Soft Methodology and Random Information Systems*, pp. 125–132. Springer. Briefly discussed in Chapter 8.
  - Antonucci, A., Salvetti, A., Zaffalon., M. (2004). Hazard assessment of debris flows by credal networks. In Pahl-Wostl, C., Schmidt, S., Rizzoli, A. E., Jakeman, A. J. (Eds), *iEMSs 2004: Complexity and Integrated Resources Management, Transactions of the 2nd Biennial Meeting of the International Environmental Modeling and Software Society*, pp. 98–103. iEMSs. Briefly discussed in Chapter 8.

- Book Chapters

  - Antonucci, A., Piatti, A., Zaffalon., M. (2007). Credal networks for hazard assessment of debris flows. In Kropp, J., Scheffran, J. (Eds), Advanced Methods for Decision Making and Risk Management in Sustainability Science. Nova Science Publishers, New York. Briefly discussed in Chapter 8.

- Papers under review

  - Antonucci, A., Zaffalon, M., de Campos, P. C. (submitted). Generalized loopy 2U: a new algorithm for approximate inference in credal networks. Briefly discussed in Section 5.2.
  - Salvetti, A., Antonucci, A., Zaffalon., M. (submitted). Spatially distributed identification of debris flow source areas by credal networks. Briefly discussed in Section 8.2.4.

In addition, the following paper has been published during my Ph.D. studies, but is not included in this thesis in order to keep its length manageable:

- Antonucci, A., Piatti, A., Zaffalon., M. (2007). Credal networks for operational risk measurement and management. In *Proceedings of the 11$^{\text{th}}$ International Conference on Knowledge-Based and Intelligent Information & Engineering Systems: KES2007*, pp. 604–611. Lectures Notes in Computer Science, Springer.

1.4 Software Issues

A number of public software tools has been employed for the numerical tests and simulations presented in this thesis. These are:

- The C++ library *SMILE* (`genie.sis.pitt.edu`), developed at the Decision Systems Laboratory of the University of Pittsburgh, has been used for Bayesian networks updating.

- The ANSI C implementation of the reverse search algorithm for vertex enumeration *LRS* (`cgm.cs.mcgill.ca/~avis/C/lrs.htm`), developed by David Avis, has been used to compute the extreme mass functions of the conditional credal sets from the probability intervals.

- A Java implementation of the L2U algorithm included in the tool *2UBayes* (`www.pmr.poli.usp.br/ltd/Software/2UBayes/2UBayes.html`), developed by Jaime Shinsuke Ide, has been used for binary credal networks updating.

- Some credal networks considered in our experiments has been generated by the generator for random Bayesian and credal networks *BNGenerator* (`www.pmr.poli.usp.br/ltd/Software/BNGenerator/index.html`), developed by Jaime Shinsuke Ide.

- GL2U-based updating has been computed by the Python/Java implementation of this algorithm (`www.idsia.ch/~sun/gl2u.html`) developed by Sun Yi.

The authors of these free software packages are gratefully acknowledged.

1.5 Notes on Style

At the very least, all of the research presented in this thesis has been done under the constant supervision of Marco Zaffalon. So I think it is justified to use the first person plural throughout the thesis. Furthermore, in the cases where the results has been done together with other persons, namely the other co-authors of the papers published during my Ph.D. studies, an explicit mention of their names is provided.

1.6 Approaching probabilistic graphical models

In the next chapter, which includes essentially background material, we review the basics of two important classes of probabilistic graphical models, namely Bayesian and credal networks. With this section we want to smooth the path of the reader in approaching the field of probabilistic graphical models and their formalism. To this end, we introduce here in a purely qualitative fashion the "philosophy" characterizing approaches based on these models.

By definition, a probabilistic graphical model defines a probability mass function over a set of variables, which are in one-to-one correspondence with the nodes of a graph. The role of the graph is to outline the conditional independencies among the variables according to specific graphical criteria. The insight there is that if two nodes are somehow separated by some other nodes according to the topology of the graph, then the conditional independendence of the corresponding variables holds. This fundamental concept, which will be formalized in the next chapter by the so-called *Markov condition*, is the key feature that allows for defining a *global* model, i.e., a model over all the variables associated to the nodes of the graph, by means of *local* probabilistic assessments, concerning only the single variables and their neighbours according to the structure of the graph. This makes the modelling phase particularly easy: the assessment of a probability mass function over many variables, whose number of joint states might be huge, does not require the modeller to explicity assess the probabilities for these joint states. These probabilities are obtained instead as a product of the probabilities assessed for the local sets of variables, according to the independence relations outlined by the graph.

The advantages of this approach are considerable also for the *inference*, i.e., when the probabilistic model is queried in order to obtain new probabilistic information about its variables. It is in fact possible to design inference algorithms that exploits the graphical structure for a more efficient computation of the inferences. The key idea there is to implement a *message propagation* scheme through the structure of the graph, that performs the computation in a distributed manner.

The findings presented in this thesis refer both to modelling issues and to inference algorithms. In fact, the new language and the corresponding equivalent representations we present in this work are based on this kind of graphical concepts, and provide a basis for the development of a new inference algorithm based on message propagation.

Chapter 2

Probabilistic Graphical Models

This is mostly a background chapter, in which the fundamentals of *Bayesian networks* (Section 2.2) and their generalization to imprecise probabilities, i.e., *credal networks* (Section 2.4) are reviewed. The generalization is obtained by means of closed convex sets of probabilities, i.e., *credal sets* (Section 2.3). We also do a short overview of the state of the art of updating algorithms for credal networks (Section 2.6). First of all, let us set up the necessary formalism.

2.1 Basic Notation

All the models we review in this chapter are based on a collection of random variables, structured as a set[1] $\mathbf{X} := \{X_1, \ldots, X_n\}$, and a *directed acyclic graph* (DAG) $\mathcal{G}$. Assume a one-to-one correspondence between the elements of $\mathbf{X}$ and the nodes of $\mathcal{G}$. Accordingly, in the following we use *node* and *variable* interchangeably. For each $X_i \in \mathbf{X}$, Π_i denotes the set of the *parents* of X_i, i.e., the random variables corresponding to the immediate predecessors of X_i according to $\mathcal{G}$. A notation with uppercase subscripts (e.g., X_E) is similarly employed to denote vectors (and sets) of variables in $\mathbf{X}$.

In our assumptions the variables in $\mathbf{X}$ take values in finite sets. For each $X_i \in \mathbf{X}$, the possibility space of X_i can be denoted as $\Omega_{X_i} := \{x_{i0}, x_{i1}, \ldots, x_{i(d_i-1)}\}$, with $d_i := |\Omega_{X_i}|$.[2] If X_i is a binary variable, the elements of Ω_i are also denoted as $\{x_i, \neg x_i\}$ in some cases and occasionally by $\{0, 1\}$. We denote by $P(X_i)$ a mass function for X_i and $P(x_i)$ the probability that $X_i = x_i$, where x_i is a generic element of Ω_{X_i}.

For both Bayesian and credal networks, we assume the *Markov condition* to make $\mathcal{G}$ represent probabilistic independence relations between the variables in

[1] The symbol := is used to denote definitions.

[2] The notation $|\Omega|$ denotes the cardinality of a set Ω.

X: every variable is independent of its non-descendant non-parents conditional on its parents. As an example, the directed acyclic graph in Figure 2.1 states that independence between X_2 and X_3 given X_1. What makes Bayesian and credal networks different is a different notion of independence and a different characterization of the conditional mass functions for each variable given the values of the parents, which are detailed respectively in Section 2.2 and Section 2.4.

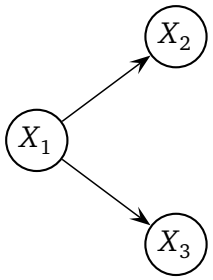


Figure 2.1: A directed acyclic graph with three nodes

All this formalism is sufficient to introduce the definition of Bayesian network, which is reviewed in the following section.

2.2 Bayesian Networks

Here we quickly review some fundamentals about Bayesian networks. For a comprehensive analysis of this topic, we point the reader to Pearl's classical textbook [Pea88].

2.2.1 Definition

Definition 1. *A Bayesian network over* $\mathbf{X}$ *is a pair* $\langle \mathcal{G}, \mathbb{P} \rangle$ *such that* $\mathbb{P}$ *is a set of conditional mass functions* $P(X_i|\pi_i)$*, one for each* $X_i \in \mathbf{X}$ *and* $\pi_i \in \Omega_{\Pi_i}$.

As noted in the previous section, we assume the *Markov condition* to make $\mathcal{G}$ represent probabilistic independence relations between the variables in $\mathbf{X}$. Thus, a Bayesian network determines a joint mass function $P(\mathbf{X})$ according to the following factorization formula:

$$P(\mathbf{x}) = \prod_{i=1}^{n} P(x_i|\pi_i), \tag{2.1}$$

for each $\mathbf{x} \in \Omega_{\mathbf{X}}$, where for each $i = 1, \ldots, n$ the values (x_i, π_i) are those consistent with $\mathbf{x}$. As an example, we can define a Bayesian network over the three binary variables (X_1, X_2, X_3) associated to the directed acyclic graph in Figure 2.1 by assigning $P(x_1) = .2$, $P(x_2|x_1) = .3$, $P(x_2|\neg x_1) = .4$, $P(x_3|x_1) = .5$ and $P(x_3|\neg x_1) = .6$. According to Equation (2.1), we have $P(x_1, x_2, x_3) = P(x_1)P(x_2|x_1)P(x_3|x_1) = .03$, and we can similarly compute the probabilities of the other seven joint states.

2.2.2 Updating

Bayesian networks can be naturally regarded as expert systems. We can query a Bayesian network to gather probabilistic information about a variable given evidence about some other variables. This task is called *updating* and consists in the computation of posterior beliefs about a queried variable X_q, given the available evidence $X_E = x_E$.

$$P(x_q|x_E) = \frac{\sum_{x_M} \prod_{i=1}^{n} P(x_i|\pi_i)}{\sum_{x_M, x_q} \prod_{i=1}^{n} P(x_i|\pi_i)}, \tag{2.2}$$

where $X_M := \mathbf{X} \setminus (\{X_q\} \cup X_E)$, the domains of the arguments of the sums are left implicit and the values of x_i and π_i are consistent with $\mathbf{x} = (x_q, x_M, x_E)$.

As an example, let us consider the updating problem consisting in the computation of $P(x_2|x_3)$ for the Bayesian network defined in the previous section. According to Equation (2.2), we have:

$$P(x_2|x_3) = \frac{\sum_{x_1} P(x_1)P(x_2|x_1)P(x_3|x_1)}{\sum_{x_1, x_2} P(x_1)P(x_2|x_1)P(x_3|x_1)} = \frac{11}{29}. \tag{2.3}$$

The evaluation of Equation (2.2) is an NP-hard task [Coo90], but in the special case of *polytrees*, Pearl's local propagation scheme allows for efficient updating [Pea88]. A polytree is a Bayesian network based on a *singly connected* directed acyclic graph, which means a graph that does not contain any undirected cycle.

Bayesian networks are powerful means to model uncertain knowledge in many situations. Nevertheless, they require *precise* probabilistic assessment, i.e., single numerical values should be provided for each conditional probability, for each node and each possible value of the parents. Some authors consider this requirement too strong, at least in some situations. Thus, we consider a possible generalization of Bayesian networks, in which closed convex sets of probability mass functions instead of single mass functions are provided. A formal description of these sets is reported in the following section.

2.3 Credal Sets

The requirement of providing *precise* probabilistic values, which is implicitly assumed for Bayesian networks, has been criticized in a number of theories. Among them, Walley's behavioural theory of *imprecise probabilities* [Wal91] provides a complete probabilistic theory based on *coherent lower previsions*, that generalizes de Finetti's classical theory based on *linear previsions* [dF74]. Remarkably, a coherent lower prevision can be equivalently expressed by (the lower envelope of) a closed convex set of linear previsions, which are in fact equivalent to (precise) probability mass functions in the case of finite supports. Accordingly, we formalize our imprecise probabilistic approaches in terms of closed convex sets of probability mass functions as stated in the following section.

2.3.1 Definition

Following Levi [Lev80], we call *credal set* a closed convex set of probability mass functions. A credal set for a random variable X is denoted by $K(X)$. We follow Cozman [Coz00] in considering only *finitely generated* credal sets, i.e., obtained as the convex hull of a finite number of mass functions for a certain variable. Geometrically, a credal set of this kind is a *polytope*. Such credal set contains an infinite number of mass functions, but only a finite number of *extreme mass functions*: those corresponding to the *vertices* of the polytope, which are in general a subset of the generating mass functions. In the following, the set of the vertices of $K(X)$ is denoted as $\text{ext}[K(X)]$. Note that there are no bounds to the possible number of vertices of a credal set, with the only exception of those over binary variables that clearly cannot have more than two extreme mass functions.

Given a non-empty subset $\Omega_X^* \subseteq \Omega_X$, an important credal set for our purposes is the *vacuous credal set* relative to Ω_X^*, i.e., the set of all the mass functions for X assigning probability one to Ω_X^*. We denote this set by $K_{\Omega_X^*}(X)$. In the following we use the well-known fact that the vertices of $K_{\Omega_X^*}(X)$ are the $|\Omega_X^*|$ degenerate mass functions assigning probability one to the single elements of Ω_X^*.

2.3.2 Inference Based on Credal Sets

For any $x \in \Omega_X$, the lower probability for x according to the credal set $K(X)$ is

$$\underline{P}^K(x) := \min_{P(X) \in K(X)} P(x). \tag{2.4}$$

If there are no ambiguities about the credal set considered in Equation (2.4), the supscript K is removed and the corresponding lower probability is simply

denoted as $\underline{P}(X)$. Similar definitions can be provided for upper probabilities, and lower and upper expectations. Walley shows that inferences based on a credal set are equivalent to those based only on its vertices [Wal91].

Given a joint credal set $K(X,Y)$, we say that X and Y are *strongly independent*, when every vertex in $K(X,Y)$ satisfies stochastic independence of X and Y. We generalize the notion of *marginalization* for probability mass functions to credal sets as follows: given a joint credal set $K_*(X,Y)$, its marginal over X is denoted by $K_*(X)$ and is obtained by the convex hull of the collection of mass functions $P_*(X)$, where, for each $P_*(X,Y) \in K_*(X,Y)$, $P_*(X)$ is obtained marginalizing over X from $P_*(X,Y)$.

Finally, regarding conditioning with credal sets, we perform elements-wise application of Bayes' rule. The posterior credal set is the union of all posterior mass functions. Denote by $K(X|Y=y)$ the set of conditional mass functions $P(X|Y=y)$ for generic variables X and Y, in this thesis we always assume non-zero lower probability for the conditioning event $(Y=y)$.

2.3.3 Credal Sets from Probability Intervals

According to the discussion in the previous section, a credal set can be specified by an explicit enumeration of probability mass functions. Alternatively we can consider a set of *probability intervals* over Ω_X, say $\mathbb{I}_X = \{\mathbb{I}_x : \mathbb{I}_x = [l_x, u_x],, 0 \leq l_x \leq u_x \leq 1, x \in \Omega_X\}$, that specifies a credal set $K(X) = \{P(X) : P(x) \in \mathbb{I}_x, x \in \Omega_X, \sum_{x\in\Omega_X} P(x) = 1\}$. $\mathbb{I}_X$ is said to *avoid sure loss* if the corresponding credal set is not empty and to be *coherent* (or *reachable*) if $u_{x'} + \sum_{x\in\Omega_X, x\neq x'} l_x \leq 1 \leq l_{x'} + \sum_{x\in\Omega_X, x\neq x'} u_x$, for all $x \in \Omega_X$. $\mathbb{I}_X$ is coherent if and only if the intervals are tight, i.e., for each lower or upper bound in $\mathbb{I}_X$ there is a mass function in the credal set at which the bound is attained [Wal91; CHM94]. Standard algorithms can compute the vertices of a credal set for which a probability interval has been provided [AF96]. Yet, the resulting number of vertices can be exponential in the input size [Tes92].

2.3.4 The Imprecise Dirichlet Model

Probability intervals can be inferred from data by the *imprecise Dirichlet model*, a generalization of Bayesian learning from i.i.d. multinomial data based on imprecise-probability modeling of prior ignorance. The bounds for the predictive probability that $X = x$ are given by

$$[\#(x)/(N+s), (\#(x)+s)/(N+s)], \tag{2.5}$$

where $\#(x)$ counts the number of units in the sample in which $X = x$, N is the total number of units, and s is a hyperparameter that expresses the degree

of caution of inferences, usually chosen in the interval $[1,2]$ (see [Wal96] for details). Note that sets of probability intervals obtained using the imprecise Dirichlet model are reachable. Some of the conditional credal sets considered by the environmental application of Chapter 8 have obtained using the imprecise Dirichlet model with $s = 2$.

2.4 Credal Networks

Credal networks generalize Bayesian networks by means of credal sets. Here we report some results and definitions related to these models. We point the reader to [Coz00] for an overview of these models, and to [Coz05] for a recent review of the state of the art in this field.

2.4.1 General Definition

Credal networks extend Bayesian nets to deal with imprecision in probability, and can be actually regarded as sets of Bayesian networks. This extension is obtained by means of the fundamental notion of credal set introduced in Section 2.3. The following definition of credal network is called *enumerative* as in fact consists in the explicit enumeration of all the Bayesian networks associated to a credal network.

Definition 2. *A credal network over* $\mathbf{X}$ *is a pair* $\langle \mathcal{G}, \{\mathbb{P}_1, \ldots, \mathbb{P}_m\}\rangle$ *such that* $\langle \mathcal{G}, \mathbb{P}_j\rangle$ *is a Bayesian network over* $\mathbf{X}$ *for each* $j = 1, \ldots, m$.

The Bayesian networks $\{\langle \mathcal{G}, \mathbb{P}_j\rangle\}_{j=1}^m$ are called to be the *compatible* Bayesian networks of the credal network specified in Definition 2.

Inferences over a credal network are intended as inferences based on a credal set for $\mathbf{X}$ determined as follows. Given the credal network $\langle \mathcal{G}, \{\mathbb{P}_1, \ldots, \mathbb{P}_m\}\rangle$, we consider the convex hull of the points $\{P_j(\mathbf{X})\}_{j=1}^m$, which are the joint mass functions determined by the compatible Bayesian networks of the credal network, i.e., [3]

$$K(\mathbf{X}) := \mathrm{CH}\{P_1(\mathbf{X}), \ldots, P_m(\mathbf{X})\}, \tag{2.6}$$

where CH is the convex hull operator. The convexification in Equation (2.6) is necessary to ensure consistency with Walley's theory of coherent lower previsions [Wal91]. With a small abuse of terminology, we call the credal set defined in Equation (2.6) the *strong extension* of the credal network, by analogy with the notion provided in the special case of *separately specified* credal networks (see

[3] Generally speaking the fact that all the joint mass functions $\{P_j(\mathbf{X})\}_{j=1}^m$ in Equation (2.6) factorize as in Equation (2.1) does not imply that the every $P(\mathbf{X}) \in K(\mathbf{X})$ should do the same.

Section 2.4.2). Inference on a credal network is intended as inference based on its strong extension, i.e., the computation of upper and lower bounds for the posterior expectation of a given function of $\mathbf{X}$, with respect to $P(\mathbf{X}) \in K(\mathbf{X})$.

2.4.2 Separately Specified Credal Networks

The main feature of probabilistic graphical models, which is the specification of a global model through a collection of sub-models local to the nodes of the graph, contrasts with Definition 2, which represents a credal network as an explicit enumeration of Bayesian networks.

Nevertheless, there are specific subclasses of credal networks that define a set of Bayesian networks as in Definition 2 through local specifications. This is for example the case of credal networks with separately specified credal sets,[4] which are simply called *separately specified credal networks* in the following. This specification requires each conditional mass function to belong to a (conditional) credal set, according to the following definition:

Definition 3. *A separately specified credal network over* $\mathbf{X}$ *is a pair* $\langle \mathcal{G}, \mathbb{K} \rangle$*, where* $\mathbb{K}$ *is a set of conditional credal sets* $K(X_i|\pi_i)$*, one for each* $X_i \in \mathbf{X}$ *and* $\pi_i \in \Omega_{\Pi_i}$*.*

According to [Coz00], the *strong extension* $K(\mathbf{X})$ of a separately specified credal network is defined as the convex hull of the joint mass functions $P(\mathbf{X})$, with, for each $\mathbf{x} \in \Omega_{\mathbf{X}}$:

$$P(\mathbf{x}) = \prod_{i=1}^{n} P(x_i|\pi_i), \qquad \begin{array}{l} P(X_i|\pi_i) \in K(X_i|\pi_i), \\ \text{for each } X_i \in \mathbf{X}, \pi_i \in \Pi_i. \end{array} \tag{2.7}$$

Here $K(X_i|\pi_i)$ can also be replaced by $\text{ext}[K(X_i|\pi_i)]$ according to the following well-known and intuitive proposition, which is proved here only because of the seemingly lack of its formal proof in the literature.

Proposition 1. *The vertices* $\{\tilde{P}_j(\mathbf{X})\}_{j=1}^{m}$ *of the strong extension* $\tilde{K}(\mathbf{X})$ *of a separately specified credal network* $\langle \mathcal{G}, \mathbb{K} \rangle$ *are joint mass functions obtained by the product of vertices of the separately specified conditional credal sets, i.e., for each* $\mathbf{x} \in \Omega_{\mathbf{X}}$*:*

$$\tilde{P}_j(\mathbf{x}) = \prod_{i=1}^{n} \tilde{P}_j(x_i|\pi_i), \tag{2.8}$$

for each $j = 1, \ldots, m$*, where, for each* $i = 1, \ldots, n$ *and* $\pi_i \in \Omega_{\Pi_i}$*,* $\tilde{P}_j(X_i|\pi_i)$ *is a vertex of* $K(X_i|\pi_i) \in \mathbb{K}$*.*

[4] Some authors use also the expression *locally defined* credal networks [Coz00].

Proof. We prove the proposition by a *reductio ad absurdum*, assuming that at least a vertex $\tilde{P}(\mathbf{X})$ of $\tilde{K}(\mathbf{X})$ is not obtained by a product of vertices of the conditional credal sets in $\mathbb{K}$. This means that, for each $\mathbf{x} \in \Omega_{\mathbf{X}}$, $\tilde{P}(\mathbf{x})$ factorizes as in Equation (2.8), but at least a conditional probability in this product comes from a conditional mass function which is not a vertex of the relative conditional credal set. This conditional mass function, say $P(X_t|\pi_t)$, can be expressed as a convex combination of vertices of $K(X_t|\pi_t)$, i.e., $P(X_t|\pi_t) = \sum_\alpha c_\alpha P_\alpha(X_t|\pi_t)$, with $\sum_\alpha c_\alpha = 1$ and, for each α, $c_\alpha \geq 0$, and $P_\alpha(X_t|\pi_t)$ is a vertex of $K(X_t|\pi_t)$. Thus, for each $\mathbf{x} \in \Omega_{\mathbf{X}}$,

$$\tilde{P}(\mathbf{x}) = \left[\sum_\alpha c_\alpha P_\alpha(x_t|\pi_t)\right] \cdot \prod_{i \neq t} P(x_i|\pi_i), \tag{2.9}$$

which can be easily reformulated as a convex combination. Thus, $\tilde{P}(\mathbf{X})$ is a convex combination of elements of the strong extension $\tilde{K}(\mathbf{X})$. This violates the assumption that $\tilde{P}(\mathbf{X})$ is a vertex of $\tilde{K}(\mathbf{X})$. □

As an example, let us define a separately specified credal network over the three binary variables (X_1, X_2, X_3) and the directed acyclic graph in Figure 2.1. In order to do that, we set $P(x_1) \in [.2, .3]$, $P(x_2|x_1) \in [.3, .4]$, $P(x_2|x_1) \in [.4, .5]$, $P(x_3|x_1) \in [.5, .6]$, $P(x_3|\neg x_1) \in [.6, .7]$. Note that, as all the variables are binary, the specification of the lower and upper bounds for the probability of the first state is a proper specification of the corresponding credal set. For instance, the two extreme mass function of the unconditional credal set $K(X_1)$ are clearly $P_1(X_1) = (.2, .8)$ and $P_2(X_1) = (.3, .7)$. Similarly, the four conditional credal sets associated to the specification of this credal network have two extreme mass functions each, and the strong extension $K(X_1, X_2, X_3)$ has therefore 32 compatible Bayesian networks corresponding to all their possible combinations.

2.4.3 Non-Separately Specified Credal Networks

Separately specified credal networks are the most popular type of credal network, but it is possible to consider credal networks that cannot be formulated as in Definition 3. This corresponds to having relationships between the different specifications of the conditional credal sets, which means that the possible values for a given conditional mass function can be affected by the values of some other conditional mass functions. A credal network of this kind is simply called *non-separately specified*.

As an example, some authors considered so-called *extensive* specifications of credal networks [FdRC02], where instead of a separate specification for each conditional mass function as in Definition 3, the *probability table* $P(X_i|\Pi_i)$, i.e.,

a function of both X_i and Π_i, is defined to belong to a finite set of tables. Figure 2.2 reports an example of an extensively specified credal network. The strong extension of an extensive credal network is obtained as in Equation (2.7), 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. Chapter 3 reports examples and motivations for non-separately specified credal networks, including also the extensive case.

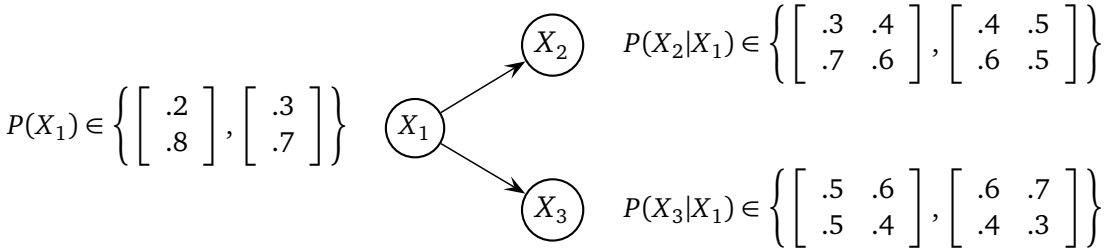


Figure 2.2: An extensive specification of a credal network over three binary variables. The compatible Bayesian networks of the credal network are those obtained by the eight possible combinations of the probability tables $P(X_2|X_1)$ and $P(X_3|X_1)$ with the two extreme mass function of $K(X_1)$. The network is non-separately specified, as the conditional mass functions over X_2, corresponding to the two columns of the conditional probability table $P(X_2|X_1)$, cannot vary independently of one other (and similarly for X_3).

2.5 Computing with Credal Networks

By an analogy with what we have done for Bayesian networks in Section 2.2.2, we can query a credal network in order to gather probabilistic information about a variable given evidence about some other variables. This task is still called *updating* and consists in the computation, with respect to the network strong extension $K(\mathbf{X})$, of $\underline{P}(x_q|x_E)$ and $\overline{P}(x_q|x_E)$. Thus, Equation (2.2) generalizes as:

$$\underline{P}(x_q|x_E) = \min_{k=1,\ldots,m} \frac{\sum_{x_M} \prod_{i=1}^{n} P_k(x_i|\pi_i)}{\sum_{x_M,x_q} \prod_{i=1}^{n} P_k(x_i|\pi_i)}, \tag{2.10}$$

and similarly with a maximum replacing the minimum for upper probabilities $\overline{P}(x_q|x_E)$. More generally, we could also be interested in the computation of the

posterior credal set for the queried variable X_q given the evidence x_E, i.e.,

$$K(X_q|x_E) := \mathrm{CH}\left\{P_k(X_q|x_E)\right\}_{k=1}^{m}. \tag{2.11}$$

Note that, according to Proposition 1, for separately specified credal networks, the number m of compatible Bayesian networks is exponential in the input size. Thus, Equation (2.10) cannot be solved in general by exhaustive iteration of updating algorithms for Bayesian networks. In fact, exact updating displays higher complexity than Bayesian networks: credal networks updating is NP-complete for polytrees[5], and $\mathrm{NP}^{\mathrm{PP}}$-complete for general credal networks [dCC05]. We point the reader to Section 2.6 for a summary about the existing algorithms for credal networks exact and approximate updating.

2.6 Algorithms for Credal Networks Updating

2.6.1 The 2U Algorithm and Its Loopy Extension

The extension to credal networks of Pearl's algorithm for efficient updating on polytree-shaped Bayesian networks faced serious computational problems. To solve Equation (2.2), Pearl's propagation scheme computes the joint probabilities $P(x_q, x_E)$ for each $x_q \in \Omega_{X_q}$; the conditional probabilities associated to $P(X_q|x_E)$ are then obtained using the normalization of this mass function. Such approach cannot be easily extended to Equation (2.10), because $\underline{P}(X_q|x_E)$ and $\overline{P}(X_q|x_E)$ are not normalized in general.

A remarkable exception to this situation is the case of *binary* credal networks, i.e., models for which all the variables are binary. The reason is that a credal set for a binary variable has at most two vertices and can therefore be identified with an interval. This enables an efficient extension of Pearl's propagation scheme. The result is an exact algorithm for polytree-shaped binary separately specified credal networks, called *2-Updating* (or simply 2U), whose computational complexity is linear in the input size. Loosely speaking, 2U computes lower and upper messages for each node according to the same propagation scheme of Pearl's algorithm but with different combination rules. Any node produces a local computation and the global computation is concluded updating all the nodes in sequence. See [FZ98] for a detailed description of 2U.

Loopy propagation is a popular technique that applies Pearl's propagation to multiply connected Bayesian networks [MWJ99]: propagation is iterated until probabilities converge or for a fixed number of iterations. In [IC04], Ide and

[5] We extend to credal networks the notion of polytree introduced for Bayesian networks in Section 2.2.2.

Cozman extend these ideas to belief updating on credal networks, by developing a loopy variant of 2U (called *loopy 2U* or simply L2U) that makes 2U usable for multiply connected binary credal networks.

Initialization of variables and messages follows the same steps used in the 2U algorithm. Then nodes are repeatedly updated following a given sequence. Updates are repeated until convergence of probabilities is observed or until a maximum number of iterations is reached. Concerning computational complexity, L2U is basically an iteration of 2U and its complexity is therefore linear in the number input size and in the number of iterations. Overall, the L2U algorithm is fast and returns good results, with low errors after a small number of iterations [IC04, Section 6]. However, at the present moment, there are no theoretical guarantees about convergence.

Briefly, L2U overcomes 2U limitations about topology, at the cost of an approximation. In Section 5.1 we show how to make it bypass also the limitations about the number of possible states.

2.6.2 Other Methods

As noted in Section 2.5, the difficulty faced by inference algorithms is due to the potentially enormous number of vertices that a strong extension may have, even for small networks. Exact inference algorithms typically examine potential vertices of the strong extension to produce the required lower/upper values [CCM94; Coz00; FdRC02; FdRC03]. Approximate inference algorithms can produce either outer or inner approximations: the former produce intervals that enclose the correct probability interval between lower and upper probabilities [CM02; dRCdC03; HDVH98; Tes92], while the latter produce intervals that are enclosed by the correct probability interval [CCM94; Coz96]. Some of these algorithms emphasize enumeration of vertices, while others resort to optimization techniques (as computation of lower/upper values for $P(x_q|x_E)$ is equivalent to minimization/maximization of a fraction containing polynomials in probability values). Rather detailed overviews of inference algorithms for imprecise probabilities have been published by Cano and Moral (e.g., [CM99]).

2.7 Summary

In this chapter the reader is given the necessary background leading up to the formalism of probabilistic graphical models. More specifically, the formal definitions of both Bayesian and credal networks are provided. The latter is a generalization to sets of probability mass function of the first. This generalization

is based on the fundamental notion of credal set, i.e., a closed convex set of probability mass functions.

This extension poses many challenges concerning both the modelling phase and the inferences. At the moment, in fact, there is no a single standard way to specify a credal network. Two main subclasses of models, called respectively separately specified and non-separately specified credal networks, exist and a different language of specification characterizes each class. Regarding inferences, those based on credal networks are considerably more difficult than those based on Bayesian networks. Despite the presence of a number of inference algorithms proposed during the last decade, there are no algorithms based on pure message-propagation schemes that can update credal networks of any kind.

The findings presented in the rest of this thesis should be regarded as an attempt to meet these challenges. More specifically, a unifying graphical language for both non-separately specified and separately specified credal networks is proposed in Chapter 4, while a message-propagation algorithm for credal networks of any kind is described in Chapter 5.

Chapter 3

Reasons for Non-Separately Specified Credal Networks: Conservative Inference Rule on Bayesian Networks and Other Problems

According to the discussion in the previous chapter, we have two different classes of credal networks: those *separately specified*, introduced in Section 2.4.2, for which each conditional mass function is allowed to vary in its credal set independently of the others, and the *non-separately specified* credal networks, that allow for relationships between conditional mass functions in different credal sets, which can be far away from each other in the net. Although the idea of non-separately specified credal nets is relatively intuitive, it should be stressed that this kind of nets has been investigated very little: in fact, there has been no attempt so far to develop a general graphical language to describe them; and there is no algorithm to compute with them.[1] This appears to be an unfortunate gap in the literature as the non-separate specification seems to be the key to model many important problems in uncertain reasoning. In this chapter, we illustrate this necessity by a few examples. An algorithmic solution for these problems are indeed provided by the theoretical results developed in Chapter 4. The first problem motivating the need of non-separately specified credal networks is an equivalence results that we prove, with respect to a specific updating problem, between credal and Bayesian networks. This is what we detail in the following section.

[1] An exception is the classification algorithm developed for the *naive credal classifier* [Zaf01], but it is ad hoc for a very specific type of network. More generally speaking, it is not unlikely that some of the existing algorithms for separately specified nets can be extended to special cases of non-separate specification, but we are not aware of any published work dealing with this issue.

3.1 Conservative Inference Rule on Bayesian Networks

In this section we establish an intimate connection between Bayesian and credal networks. We focus on traditional belief updating with credal networks, and on the kind of belief updating that arises with Bayesian networks when the reason for the missingness of some of the unobserved variables in the network is unknown. We show that the two updating problems are formally the same. Notably, in order to obtain the equivalence, also non-separately specified credal networks should be considered.

Imagine the following situation. You want to use a graphical model to formalize your uncertainty about a domain. You prefer precise probabilistic models and so you choose Bayesian networks. You take care to precisely specify the graph and all the conditional mass functions required. At this point you are done with the modeling phase, and start updating beliefs about a target variable conditional on the observation of some variables in the net. The remaining variables are not observed, i.e., they are *missing*. You know that some of the missing variables are simply *missing at random* (MAR, see [LR87]), and so they can easily be dealt with by traditional approaches. Yet, there is a subset of missing variables for which you do not know the process originating the missingness.

This innocuous-looking detail is going to change the very nature of your model: while you think you are working with Bayesian networks, what you are actually using are credal networks.

The implicit passage from Bayesian to credal nets is based on two steps. First, the above conditions, together with relatively weak assumptions, give rise to a specific way to update beliefs called *conservative inference rule* (CIR, see Section 3.1.1) [Zaf05]. CIR is an imprecise-probability rule: it leads, in general, to imprecise posterior probabilities for the target variable, even if the original model is precise. The second step is done in Section 3.1.2: we show the formal equivalence between CIR-based updating in Bayesian networks, and the traditional credal-network updating described in Section 2.5.

CIR and credal networks have been proposed with quite different motivations in the literature: CIR as an updating rule for the case of partial ignorance about the missingness (or incompleteness) process; credal networks as a way to relax the strict modeling requirements imposed by precise graphical models. The main interest in our result is just the established connection between two such seemingly different worlds. But the result appears also to be a basis for using algorithms for credal networks to solve CIR-based updating problems (as in fact we do in Section 4.4).

3.1.1 Conservative Inference Rule

The most popular approach to missing data in the literature and in the statistical practice is based on the so-called *missing-at-random* assumption [LR87]. MAR allows missing data to be neglected, thus turning the incomplete data problem into one of complete data. Unfortunately, MAR embodies the idea that the process responsible for the missingness (i.e., the *missingness process*) is not selective, which is not realistic in many cases. 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 [dCZ04]. This result has been expanded by Zaffalon [Zaf05] 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 credal set as our updated beliefs about the queried variable:

$$K(X_q||^{X_I}x_E) := \mathrm{CH}\left\{P(X_q|x_E,x_I)\right\}_{x_I\in\Omega_{X_I}}, \tag{3.1}$$

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 clearly $P(X_q|x_E,x_I)=\sum_{x_M}P(X_q,x_M|x_E,x_I)$.

3.1.2 Equivalence between CIR-Based Updating in Bayesian Nets and Credal Nets Updating

In this section we prove the formal equivalence between updating with CIR on Bayesian networks and standard updating on credal networks, defining two distinct mappings from a generic instance of the first problem in a corresponding instance of the second and *vice versa*. Figure 3.1 reports the correspondence scheme with the names of the mappings that will be introduced next. We focus on the case of Bayesian networks assigning positive probability to each event.

From Bayesian to credal networks

First let us define the B2C transformation, mapping a Bayesian network $\langle\mathcal{G},\mathbb{P}\rangle$, where a subset X_I of $\mathbf{X}$ is specified, in a credal network. For each variable $X\in$

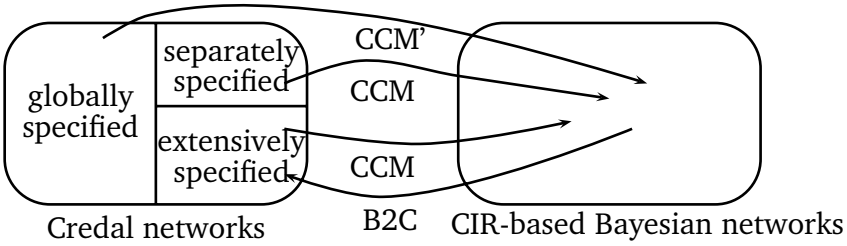


Figure 3.1: Relations between updating on credal networks and CIR-updating in Bayesian networks.

X_I, B2C prescribes to: (i) add to X an *auxiliary child node*[2] X', associated to a binary variable with possible values x' and $\neg x'$; and (ii) extensively specify the probability table $P(X'|X)$, to belong to the following set of $|\Omega_X|$ tables:

$$\left\{ \begin{bmatrix} 100\ldots0 \\ 011\ldots1 \end{bmatrix}, \ldots, \begin{bmatrix} 0\ldots010\ldots0 \\ 1\ldots101\ldots1 \end{bmatrix}, \ldots, \begin{bmatrix} 000\ldots01 \\ 111\ldots10 \end{bmatrix} \right\}. \tag{3.2}$$

Each table in Eq. (3.2) specifies a conditional probability for the state x' of X' (corresponding to the first row of the table), which is zero conditionally on any state of X except a single one, different for any table. The B2C transformation, clearly linear in the input size, is the basis for the following:

Theorem 1. *Consider a CIR instance on a Bayesian network $\langle \mathcal{G}, \mathbb{P} \rangle$ over $\mathbf{X}$. Let $X_I \subset \mathbf{X}$ be the array of the unobserved non-MAR variables. Let $K(X_q||^{X_I} x_E)$ be the credal set returned by CIR for a queried variable X_q given the evidence $X_E = x_E$. If $K(X_q|x_E, x'_I)$ is the posterior credal set for X_q in the credal network $\langle \mathcal{G}', \mathbb{P}'_1, \ldots, \mathbb{P}'_m \rangle$ over $\mathbf{X} \cup X'_I$, obtained from $\langle \mathcal{G}, \mathbb{P} \rangle$ by a B2C transformation with the nodes X_I specified, conditional on the evidences $X_E = x_E$ and $X'_I = x'_I$, then:*[3]

$$K(X_q||^{X_I} x_E) = K(X_q|x_E, x'_I). \tag{3.3}$$

Proof. According to Eq. (3.1) and Eq. (2.11) respectively, we have:

$$K(X_q||^{X_I} x_E) = \mathrm{CH}\{P(X_q|x_E, \tilde{x}_I)\}_{\tilde{x}_I \in \Omega_{X_I}} \tag{3.4}$$

$$K(X_q|x_E, x'_I) = \mathrm{CH}\{P'_k(X_q|x_E, x'_I)\}_{k=1}^{m}. \tag{3.5}$$

[2]This transformation is inspired by Pearl's prescriptions about boundary conditions for propagation [Pea88, Section 4.3].

[3]Theorem 1 can be extended also to CIR instances modeling incomplete observations where the value of the observed variable is know to belong to a generic subset of the possibility space, rather than missing observations for which the universal space is considered.

An obvious isomorphism holds between $\{P'_k(\mathbf{X}')\}_{k=1}^m$ and Ω_{X_I}: that follows from the correspondence, for each $X_i \in X_I$, between the conditional probability tables for $P(X'_i|X_i)$ as in Eq. (3.2) and the elements of Ω_{X_i}. Accordingly, we denote by $\tilde{x}_I$ the element of Ω_{X_I} corresponding to $P'_k(\mathbf{X}')$. The thesis is proved by showing, for each $k = 1, \ldots, m$, $P'_k(X_q|x_E) = P(X_q|x_E, \tilde{x}_I)$. For each $x_q \in \Omega_{X_q}$:

$$P(x_q|x_E, \tilde{x}_I) = \sum_{x_M} P(x_q, x_M|x_E, \tilde{x}_I) \propto \sum_{x_M} P(x_q, x_M, x_E, \tilde{x}_I) \tag{3.6}$$

$$P'_k(x_q|x_E, x'_I) = \sum_{x_M, x_I} P'_k(x_q, x_M, x_I|x_E, x'_I) \propto \sum_{x_M, x_I} P'_k(x_q, x_M, x_I, x_E, x'_I) \tag{3.7}$$

According to the Markov condition:

$$P'_k(x_q, x_M, x_I, x_E, x'_I) = \prod_{i:X_i \in X_I} \left[P'_k(x'_i|x_i) \cdot P'_k(x_i|\pi_i) \right] \cdot \prod_{j:X_j \in \vec{X}' \setminus (X_I \cup X'_I)} P'_k(x_j|\pi_j), \tag{3.8}$$

with the values of x'_i, x_i, π_i, x_j and π_j consistent with $(x_q, x_M, x_E, x_I, x'_I)$.

According to Eq. (3.2), $P(x'_i|x_i)$ is zero for each $x_i \in \Omega_{X_i}$ except for the value $\tilde{x}_i$, for which is one. The sum over $x_i \in \Omega_{X_i}$ of the probabilities in Equation (3.8) is therefore reduced to a single non-zero term. Thus, taking all the sums over X_i with $X_i \in X_I$:

$$\sum_{x_I} P'_k(x_q, x_M, x_I, x_E, x'_I) = \prod_{i:X_i \in X_I} P(\tilde{x}_i|\pi_i) \cdot \prod_{j:X_j \in \vec{X} \setminus X_I} P(x_j|\pi_j) = P(x_q, x_M, x_E, \tilde{x}_I), \tag{3.9}$$

with the values of π_i, x_j and π_j consistent with $(x_q, x_M, x_E, \tilde{x}_I)$. But Equation (3.9) allows us to rewrite Equation (3.6) as Equation (3.7) and conclude the thesis. □

From credal to Bayesian networks

For credal networks specified as in Definition 2, we define a transformation that returns a Bayesian network given a credal network as follows. The Bayesian network is obtained: (i) adding a *dummy* node X'' that is parent of all the nodes in $\mathbf{X}$ (see Figure 3.2 left) and such that there is a one-to-one correspondence between the elements of $\Omega_{X''}$ and those of $\{P'_k(\mathbf{X})\}_{k=1}^m$; and (ii) setting for each $X_i \in \mathbf{X}$ and $x'' \in \Omega_{X''}$: $P(X_i|\Pi_i, x'') := P'_k(X_i|\Pi_i)$, where Π_i are the parents of X_i in the credal network and $P'_k(\mathbf{X})$ is the element of $\{P'_k(\mathbf{X})\}_{k=1}^m$ corresponding to x''.

In the special case of extensively specified credal networks, we consider a slightly different transformation, where: (i) we add a dummy node X''_i for each $X_i \in \mathbf{X}$, that is parent only of X_i (see Figure 3.2 right) and such that there is

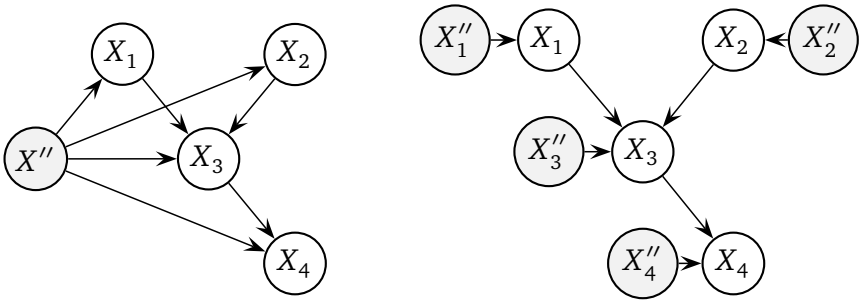


Figure 3.2: The Bayesian networks returned by CCM' (left) and CCM (right). Dummy nodes are gray, while the nodes of the original credal network are white.

a one-to-one correspondence between the elements of $\Omega_{X''_i}$ and the probability tables $P(X_i|\Pi_i)$ in the extensive[4] specification of $K(X_i|\Pi_i)$; and (ii) we set for each $X_i \in \mathbf{X}$: $P(X_i|\Pi_i, x''_i) := P'_k(X_i|\Pi_i)$, where Π_i are the parents of X_i in the credal network and $P'_k(X_i|\Pi_i)$ is the probability table of $K(X_i|\Pi_i)$ relative to x''_i. Note that no prescriptions are given about the unconditional mass functions for the dummy nodes in both the transformations, because irrelevant for the results we obtain. The second is the so-called CCM transformation [CCM94] for credal networks, while the first is simply an extension of CCM to the case of globally specified credal networks and will be denoted as CCM'. These transformations are the basis for the following:

Theorem 2. *Let $K(X_q|x_E)$ be the posterior credal set of a queried variable X_q, given some evidence $X_E = x_E$, for a credal network $\langle \mathcal{G}, \mathbb{P}_1, \ldots, \mathbb{P}_m \rangle$. Let also $\langle \mathcal{G}', \mathbb{P}' \rangle$ be the corresponding Bayesian network obtained through CCM' (or CCM if the credal network is not globally specified). Denote as $K(X_q||^{X''}x_E)$ the CIR-based posterior credal set for X_q in the Bayesian network obtained assuming what follows: the nodes in X_E instantiated to the values x_E, the dummy nodes, denoted as X'' also if CCM is used, to be not-MAR and the remaining nodes MAR. Then:*

$$K(X_q|x_E) = K(X_q||^{X''}x_E). \tag{3.10}$$

Proof. Consider a credal network specified as in Definition 2, for which CCM' should be used and X'' denotes a single dummy node. According to Equation (3.1):

$$K(X_q||^{X''}x_E) = \mathrm{CH}\{P(X_q|x_E, x'')\}_{x'' \in \Omega_{X''}}. \tag{3.11}$$

[4]Separately specified credal sets can be extensively specified, considering all the probability tables obtained from the combinations of the vertices of the original credal sets. Although correct, this transformation gives rise to an exponential explosion of the number of tables.

Setting $X_M := \mathbf{X} \setminus (X_E \cup \{X_q\})$, for each $x_q \in \Omega_{X_q}$:

$$P(x_q|x_E, x'') = \sum_{x_M} P(x_q, x_M|x_E, x'') \propto \sum_{x_M} P(x_q, x_M, x_E, x''). \tag{3.12}$$

According to the Markov condition and CCM' definition, we have:

$$P(x_q, x_M, x_E, x'') = P(x'') \cdot \prod_{i=1}^{n} P(x_i|\pi_i, x'') \propto \prod_{i=1}^{n} \tilde{P}(x_i|\pi_i) = P'_k(x_q, x_M, x_E), \tag{3.13}$$

where $\tilde{P}(\mathbf{X})$ is the joint mass function corresponding to the compatible Bayesian network associated to $x'' \in \Omega_{X''}$. The sum over x_M of the probabilities in Equation (3.13) is proportional to $\tilde{P}(x_q|x_E)$. Thus, $\tilde{P}(X_q|x_E) = P(X_q|x_E, x'')$ for each $(\tilde{P}, x'')$, that proves the thesis. Analogous considerations can be done for extensive and separate specification of credal networks transformed by CCM. □

3.1.3 Comments

We have proved the formal equivalence between two updating problems on different graphical models: CIR-based updating on Bayesian networks and traditional updating with credal networks. The result follows easily via simple transformations of the graphical models. An important consequence of the established link between Bayesian networks and credal networks is that under realistic conditions of partial ignorance about the missingness process, working with Bayesian networks is actually equivalent to working with credal networks. This appears to make credal networks even more worthy of investigation than before.

Here we have mapped CIR problems on Bayesian networks to standard updating on extensively specified credal networks, while the existing algorithms for credal networks consider the case of separately specified credal networks. This limitation is overcome in Section 4.4.1, where our result, together with the formalism developed in Section 4.1, is employed to develop a first algorithm for CIR-based updating on Bayesian networks.

3.2 Qualitative Networks

Qualitative probabilistic networks [Wel90] can be regarded as an abstraction of Bayesian networks, where the probabilistic assessments are replaced by qualitative relations describing the influences or synergies between the variables. If we regard qualitative nets as credal nets, we see that not all types of relations can

be represented by separate specifications of the conditional credal sets. This is, for instance, the case of (positive) *qualitative influence*, which requires, for two binary variables A and B, that

$$P(a|b) \geq P(a|\neg b). \tag{3.14}$$

The qualitative influence between A and B can therefore be modeled by requiring $P(A|b)$ and $P(A|\neg b)$ to belong to credal sets, which cannot be separately specified because of the constraint in Equation (3.14). An extensive specification for A should therefore be considered to model the positive influence of B [CdCIFdR04].

3.3 Equivalent Graphs for Credal Networks

Remember that DAGs represent independencies between variables according to the Markov condition. Different DAGs describing the same independencies are called *equivalent* [VP91]. Thus, a Bayesian network can be reformulated using an equivalent DAG. The same holds with credal networks, when (as implicitly done in this thesis) *strong independence* replaces standard probabilistic independence in the Markov condition [MC02].

Consider, for example, $A \to B$ and $B \to A$, which are clearly equivalent DAGs. One problem with separately specified credal networks is that they are not closed under this kind of (equivalent) structure changes: if we define a separately specified credal network for $A \to B$, and then reverse the arc, the resulting net is not separately specified in general.

In order to see that, we consider the following specification of a credal network over $A \to B$, where both A and B are binary variables: $\frac{1}{4} \leq P(a) \leq \frac{1}{2}$, $\frac{1}{2} \leq P(b|a) \leq \frac{3}{4}$ and $P(b|\neg a) = \frac{3}{4}$. As all the variables are binary, the computation of the credal set corresponding to these intervals is trivial. E.g., the vertices of $K(A)$ are clearly the two mass functions $[\frac{1}{4}, \frac{3}{4}]^T$ and $[\frac{1}{2}, \frac{1}{2}]^T$. Overall, we have a separately specified credal network with four compatible Bayesian networks, corresponding to the possible combinations of the two vertices of $K(A)$ with the two vertices of $K(B|a)$. From the joint mass functions corresponding to these Bayesian networks, we can evaluate the conditional mass functions for the corresponding Bayesian networks over $B \to A$, which are those corresponding to the following probabilities:

$$\begin{array}{llll}
P_1(b) = \frac{11}{16} & P_2(b) = \frac{3}{4} & P_3(b) = \frac{5}{8} & P_4(b) = \frac{3}{4} \\
P_1(a|b) = \frac{2}{11} & P_2(a|b) = \frac{1}{4} & P_3(a|b) = \frac{2}{5} & P_4(a|b) = \frac{1}{2} \\
P_1(a|\neg b) = \frac{2}{5} & P_2(a|\neg b) = \frac{1}{4} & P_3(a|\neg b) = \frac{2}{3} & P_4(a|\neg b) = \frac{1}{2}.
\end{array}$$

According to Definition 2, these four distinct Bayesian network specifications define a credal network over $B \to A$, which cannot be separately specified as in Definition 3. To see this, note for example that the specification $P(b) = \frac{5}{8}$, $P(a|b) = \frac{1}{2}$ and $P(a|\neg b) = \frac{2}{3}$, which would be possible if the conditional credal sets were separately specified, leads to the inadmissible value $P(a) = \frac{9}{16} > \frac{1}{2}$.

It is useful to observe that general, non-separately specified, credal networks do not suffer for these problems just because of their definition.

3.4 Learning from Incomplete Data

Given three binary random variables A, B and C, let the DAG $A \to B \to C$ express independencies between them. We want to learn the model probabilities (i.e., the parameters) for such a DAG from the incomplete data set in Table 3.1, assuming no information about the process making the observation of B missing in the last record of the data set. The most conservative approach in this case is to learn two distinct Bayesian networks from the two complete data sets corresponding to the possible values of the missing observation, and consider indeed the credal network made of these compatible Bayesian networks.

A	B	C
a	b	c
$\neg a$	$\neg b$	c
a	b	$\neg c$
a	$*$	c

Table 3.1: A data set about three binary variables; "$*$" 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.[5] Let $P_1(A,B,C)$ and $P_2(A,B,C)$ be the joint mass functions obtained in this way, which define the same conditional mass functions for

$$P_1(a) = P_2(a) = \tfrac{3}{4}$$
$$P_1(b|\neg a) = P_2(b|\neg a) = 0$$
$$P_1(c|\neg b) = P_2(c|\neg b) = 1;$$

[5] We do this only for illustrative purposes, as there are arguably better ways to learn probabilities from data, such as the *imprecise Dirichlet model* [Wal96]. Yet, also these other methods would incur the same problem [CGOM07].

and different conditional mass functions for

$$P_1(b|a) = 1 \qquad P_2(b|a) = \frac{2}{3}$$
$$P_1(c|b) = \frac{2}{3} \qquad P_2(c|b) = \frac{1}{2}.$$

We have therefore obtained two Bayesian networks over $A \rightarrow B \rightarrow C$, which can be regarded as the compatible Bayesian networks of a credal network. Such a credal network is non-separately specified. To see that, just note that if the credal network would be separately specified the values $P(b|a) = 1$ and $P(c|b) = \frac{1}{2}$ 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.

3.5 Summary and Outlooks

The need of a general formalism together with a corpus of inference algorithms for non-separately specified credal networks has been advocated by means of four important problems of uncertain reasoning.

First, we have determined a one-to-one correspondence between Bayesian networks updating based on *conservative inference rule* and standard updating on credal networks. Notably, the first problem can be mapped into the latter, only if we consider also non-separately specified credal networks.

We have also shown that the kind of constraints between conditional probabilities assumed by qualitative networks can be regarded as non-separate constraints between the different conditional credal sets. Furthermore, we have illustrated by an example that the class of credal networks is closed under the transformation of their DAGs into DAGs expressing the same dependence relations only if we consider also non-separately specified models. Finally, the quantification of a credal network from incomplete datasets has been proved to require, in general, a non-separate specification of its conditional credal sets.

With respect to future reasearch, it seems possible to extend the class of updating problem on Bayesian (and also credal) networks with missing data, that can be mapped into standard updating problem on non-separately specified credal networks. More specifically, the natural development of the transformation detailed in Section 3.1.2, would concern the case of soft evidence specified by a collection of likelihood ratio, for which a generalization to imprecise probabilities seems to be possible. The mapping into a standard updating problem on a credal network would represent therefore a generalization of Pearl's virtual evidence method to sets of probability mass functions.

Chapter 4

Decision-Theoretic Specification of Credal Networks: A Unified Language for Uncertain Modeling with Sets of Bayesian Networks

According to the discussion in the previous chapter, there are a number of reasons for which both separately and non-separately specified credal networks should be considered. An important question is whether or not all those credal networks can be represented in a way that emphasizes locality. The answer is clearly positive for separately specified credal networks. In fact, for these models, each conditional mass function is allowed to vary in its credal set independently of the others. The representation is naturally local because there are no relationships between different credal sets. The question is more complicated for non-separately specified credal networks, which can be formulated only by the enumerative specification in Definition 2 and not as in Definition 3. The idea of non-separately specified credal nets is in fact to allow for relationships between conditional mass functions in different credal sets, which can be far away from each other in the net.

In this chapter we give two major contributions. First, we define a unified graphical language to locally specify credal networks in the general case (Section 4.1). This specification is called *decision-theoretic* being inspired, via the Cano-Cano-Moral (CCM) transformation [CCM94], by the formalism of *influence diagrams*, and more generally of *decision graphs* [ZQP93]. In this language the graph of a credal network is augmented with control nodes that express the relationships between different credal sets. We give examples to show that the new language provides one with a natural way to define non-separately specified nets; and we give a procedure to reformulate any separately specified net in

the new language.

Second, we make a very simple observation (Section 4.2), which has surprisingly powerful implications: we show that for any credal network specified with the new language there is a separately specified credal network, defined over a larger domain, which is equivalent. The procedure to transform the former into the latter network is very simple, and takes only linear time. The key point is that this procedure can be used as a tool to "separate" the credal sets of non-separately specified nets. This makes it possible to model, by separately specified nets, problems formerly modeled by non-separately specified ones; and hence to use *any* (both exact and approximate) existing algorithm for separately specified nets to solve such problems.

In Section 4.3 we explore this possibility in the case of the 2U algorithm.We show that the algorithm, originally designed only for separately specified credal networks, can be extended to deal exactly and efficiently also with a class of non-separately specified models.

Our contributions also apply to the problem of belief updating on Bayesian networks by the conservative inference rule. In Section 3.1.2, this problem has been mapped onto a standard updating problem on a non-separately specified credal network, a result not straightforward to exploit in practice because of the lack of algorithms for non-separately specified credal networks. A feasible solution of this problem based on our formalism is presented in Section 4.4. First, we represent the problem by the new decision-theoretic language. Second, we use our transformation to reformulate the problem on a separately specified credal network defined over a larger domain. At this point, the problem can be solved by the existing algorithms for separately specified credal nets. Additionally, we also prove the NP-hardness of belief updating with this rule by similar transformations based on the results presented in this chapter.

4.1 Decision-Theoretic Specification of Credal Networks

In this section we provide an alternative definition of credal network that can be employed for both non-separately and separately specified credal networks. In rder to outline the main idea of our approach, let us consider the two small examples of credal networks over two binary variables reported respectively in Figure X.a and b. In both the cases, we augment the network with a binary node D_1, whose two possible values are used to enumerate the two extreme mass function of the credal set $K(X_1)$. Similarly, for the credal network in Figure X, whose conditional credal sets $K(X_2|x_1)$ and $K(X_2|\neg x_1)$ are not separately specified, we use the node D_2, whose two possible values are used to enumerate the two possible specification of the conditional probability tables. On the other

side, in the case of the credal network in Figure Xb, we use the decision node D_3, whose three possible values index the two extreme mass function of $K(X_2|x_1)$ and the two extreme mass function of $K(X_2|\neg x_1)$. Finally, in order to describe the fact that each conditional credal set has different sets of probability mass function, we just assume that the possible values of D_3 could be different for different values of X_1, as.

We provide an alternative definition of credal network with the same generality of Definition 2, but obtained through local specifications as in Definition 3. This result, which is inspired by the formalism of *decision networks* [ZQP93] via the CCM transform [CCM94], is reported in Section 4.1.1.

Remarkably, both non-separately (Section 4.1.2) and separately specified credal networks (Section 4.1.3) can be reformulated in accord to this definition by means of transformations taking only polynomial time. We can therefore regard the new definition as the basis for a graphical language to represent in a unified form credal networks of any kind.

4.1.1 General Definition of Decision-Theoretic Specification

Definition 4. *A decision-theoretic specification of a credal network over* $\mathbf{X}$ *is a triplet* $\langle \mathcal{G}', \mathbb{O}, \mathbb{P}' \rangle$ *such that: (i)* $\mathcal{G}'$ *is a DAG over* $\mathbf{X}' := (\mathbf{X}_D, \mathbf{X})$*; (ii)* $\mathbb{O}$ *is a collection of non-empty sets* $\Omega_{X_i}^{\pi_i} \subseteq \Omega_{X_i}$*, one for each* $X_i \in \mathbf{X}_D$ *and* $\pi_i \in \Omega_{\Pi_i}$*;*[1] *(iii)* $\mathbb{P}'$ *is a set of conditional mass functions* $P'(X_j|\pi_j)$*, one for each* $X_j \in \mathbf{X}$ *and* $\pi_j \in \Omega_{\Pi_j}$*.*

We intend to show that Definition 4 specifies a credal network over the variables in $\mathbf{X}$; the nodes corresponding to $\mathbf{X}$ are therefore called *uncertain* and will be displayed by circles, while those corresponding to $\mathbf{X}_D$ are called *decision nodes* and will be displayed by squares. Let us associate each decision node $X_i \in \mathbf{X}_D$ with its collection of so-called *decision functions*. For each $X_i \in \mathbf{X}_D$, the decision functions of X_i are all the possible maps $f_{X_i} : \Omega_{\Pi_i} \to \Omega_{X_i}$ returning an element of $\Omega_{X_i}^{\pi_i}$ for each $\pi_i \in \Omega_{\Pi_i}$. Note that the decision functions of a root node X_i are the single elements of Ω_{X_i}. Call *strategy* $\mathbf{s}$ an array of decision functions, one for each $X_i \in \mathbf{X}_D$. We denote as $\Omega_{\mathbf{S}}$ the set of all the possible strategies.

Each strategy $\mathbf{s} \in \Omega_{\mathbf{S}}$ determines a Bayesian network over $\mathbf{X}'$ via Definition 4, as illustrated below. A conditional mass function $P'(X_j|\pi_j)$ for each uncertain node $X_j \in \mathbf{X}$ and $\pi_j \in \Omega_{\Pi_j}$ is already specified by $\mathbb{P}'$. To determine a Bayesian network we have then to simply represent decision functions by mass functions: for each decision node $X_i \in \mathbf{X}_D$ and $\pi_i \in \Omega_{\Pi_i}$, we consider the conditional mass function $P'_{\mathbf{s}}(X_i|\pi_i)$ assigning all the mass to the value $f_{X_i}(\pi_i) \in \Omega_{X_i}$, where f_{X_i} is the decision function corresponding to $\mathbf{s}$. The Bayesian network obtained in this

[1] If X_i corresponds to a root node of $\mathcal{G}$, a single set equal to the whole Ω_{X_i} is considered.

way will be denoted as $\langle \mathcal{G}', \mathbb{P}'_{\mathbf{s}} \rangle$, while for the corresponding joint mass function, we clearly have, for each $\mathbf{x}' = (\mathbf{x}_D, \mathbf{x}) \in \Omega_{\mathbf{X}'}$, the following factorization:

$$P'_{\mathbf{s}}(\mathbf{x}_D, \mathbf{x}) = \prod_{X_i \in \mathbf{X}_D} P'_{\mathbf{s}}(x_i|\pi_i) \cdot \prod_{X_j \in \mathbf{X}} P'(x_j|\pi_j). \tag{4.1}$$

The next step is then obvious: we want to define a credal network over $\mathbf{X}$ by means of the set of Bayesian networks determined by all the possible strategies $\mathbf{s} \in \Omega_{\mathbf{S}}$. The question, at this point, is whether or not all these networks have the same DAG, as required by Definition 2. To show this we need to introduce the following transformation that removes from $\mathcal{G}'$ the decision nodes by maintaining the dependence relations between the other nodes:

Transformation 1. *Given a decision-theoretic specification of a credal network $\langle \mathcal{G}', \mathbb{O}, \mathbb{P}' \rangle$, obtain a DAG $\mathcal{G}$ associated to the variables $\mathbf{X}$ iterating, for each decision node $X_i \in \mathbf{X}_D$, the following operations over $\mathcal{G}'$: (i) draw an arc from each parent of X_i to each child of X_i; (ii) remove the node X_i.*

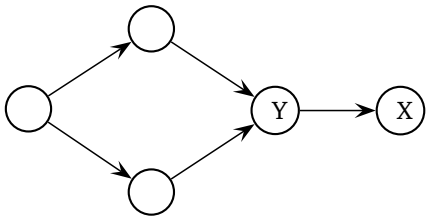


Figure 4.1: The DAG $\mathcal{G}$ returned by Transf. 1 given a decision-theoretic specification of a credal network whose DAG is that in Figure 4.2 (or also Figure 4.3 or Figure 4.4 or Figure 4.5).

Figure 4.1 reports an example of the output of Transformation 1. The DAG $\mathcal{G}$ returned by Transformation 1 is considered by the next theorem.

Theorem 3. *The marginal for $\mathbf{X}$ relative to $\langle \mathcal{G}', \mathbb{P}'_{\mathbf{s}} \rangle$, i.e., the mass function $P_{\mathbf{s}}(\mathbf{X})$ such that*

$$P_{\mathbf{s}}(\mathbf{x}) := \sum_{\mathbf{x}_D \in \Omega_{\mathbf{X}_D}} P'_{\mathbf{s}}(\mathbf{x}_D, \mathbf{x}), \tag{4.2}$$

for each $\mathbf{x} \in \Omega_{\mathbf{X}}$, factorizes as the joint mass function of a Bayesian network $\langle \mathcal{G}, \mathbb{P}_{\mathbf{s}} \rangle$ over $\mathbf{X}$, where $\mathcal{G}$ is the DAG obtained from $\mathcal{G}'$ by Transformation 1.

Proof. Let us start the marginalization in Equation (4.2) from a decision node $X_j \in \mathbf{X}_D$. According to Equation (4.1), for each $\mathbf{x}' \in \Omega_{\mathbf{X}'}$:

$$\sum_{\mathbf{x}_j \in \Omega_{\mathbf{X}_j}} P'_{\mathbf{s}}(\mathbf{x}') = \sum_{\mathbf{x}_j \in \Omega_{\mathbf{X}_j}} \left[\prod_{X_l \in \mathbf{X}_D} P'_{\mathbf{s}}(x_l|\pi_l) \cdot \prod_{X_i \in \mathbf{X}} P'(x_i|\pi_i) \right]. \tag{4.3}$$

Thus, moving out of the sum the conditional probabilities that do not refer to the states of X_j (which are briefly denoted by Δ), Equation (4.3) becomes:

$$\Delta \cdot \sum_{x_j \in \Omega_{X_j}} \left[P'_{\mathbf{s}}(x_j|\pi_j) \cdot \prod_{X_r \in \Gamma_{X_j}} P'(x_r|x_j, \tilde{\pi}_r) \right], \tag{4.4}$$

where Γ_{X_j} denotes the children of X_j and, for each $X_r \in \Gamma_{X_j}$, $\tilde{\Pi}_r$ are the parents of X_r deprived of X_j. Therefore, considering that the mass function $P'_{\mathbf{s}}(X_j|\pi_j)$ assigns all the mass to the value $f_{X_j}(\pi_j) \in \Omega_{X_j}$, where f_{X_j} is the decision function associated to $\mathbf{s}$, Equation (4.4) rewrites as

$$\Delta \cdot \prod_{X_r \in \Gamma_{X_j}} P'(x_r|f_{X_j}(\pi_j), \tilde{\pi}_r). \tag{4.5}$$

It is therefore sufficient to set $\overline{\Pi}_r := \Pi_j \cup \tilde{\Pi}_r$, and

$$P_{\mathbf{s}}(X_r|\overline{\pi}_r) := P'(X_r|f_{X_j}(\pi_j), \tilde{\pi}_r), \tag{4.6}$$

to regard Equation (4.5) as the joint mass function of a Bayesian network over $\mathbf{X}' \setminus \{X_j\}$ based on the DAG returned by Transformation 1 considered for the single decision node $X_j \in \mathbf{X}_D$. The thesis therefore follows from a simple iteration over all the $X_j \in \mathbf{X}_D$. □

From this, considering the Bayesian networks $\langle \mathcal{G}, \mathbb{P}_{\mathbf{s}} \rangle$ for each strategy $\mathbf{s} \in \Omega_{\mathbf{S}}$ as compatible Bayesian networks of a credal network, it is straightforward to obtain the following result:

Corollary 1. *A decision-theoretic specification of a credal network as in Definition 4 defines a credal network over* $\mathbf{X}$*, based on the DAG* $\mathcal{G}$ *returned by Transformation 1.*

The strong extension of $\langle \mathcal{G}', \mathbb{O}, \mathbb{P}' \rangle$ is therefore intended as the strong extension $K(\mathbf{X})$ of the credal network considered in Corollary 1. What we show in the next sections is how to provide decision-theoretic specifications of credal networks, according to Definition 4, for both separately and non-separately specified credal networks.

4.1.2 Decision-Theoretic Specification of Non-Separately Specified Credal Networks

It is worth to note that any credal network defined as in Definition 2 can be reformulated as in Definition 4, by simply adding a single decision node, which is parent of all the other nodes (see Figure 4.2).

The conditional mass functions, corresponding to different values of the decision node, are assumed to be those specified by the compatible Bayesian networks. This means that, if D denotes the decision node, the states of D index the compatible Bayesian networks, and $P(X_i|\pi_i, d) := P_d(X_i|\pi_i)$, where $P_d(X_i|\pi_i)$ are the conditional mass functions specified by the d-th compatible Bayesian network for each $X_i \in \mathbf{X}$ and $\pi_i \in \Omega_{\Pi_i}$ and $d \in \Omega_D$. This formulation, which is an example of the CCM transformation [CCM94], is only seemingly local, because of the arcs connecting the decision node with all the uncertain nodes. However, in many cases, this is not the only way to provide a decision-theoretic specification of a credal network.

Consider, for example, the class of extensively specified credal networks introduced in Section 2.4.3. We can provide a decision-theoretic specification, as in Definition 4, of a credal network of this kind by introducing a decision parent for each node of the original credal network (Figure 4.3). The conditional mass functions of the uncertain nodes corresponding to different values of the related decision nodes are assumed to be those specified by the different tables in the extensive specification. This means that, if X_i is an uncertain node and D_i the corresponding decision node, the states $d_i \in \Omega_{D_i}$ index the tables $P_{d_i}(X_i|\Pi_i)$ of the extensive specification for X_i, and, for each $\pi_i \in \Pi_i$, $P(X_i|d_i, \pi_i)$ is the mass function $P_{d_i}(X_i|\pi_i)$ associated to the d_i-th table of the extensive specification. E.g., the two tables defined in the extensive specification of the node B for the credal network in Figure 2.2, can be indexed by a binary decision parent D:

$$P(B|a, d) = \begin{bmatrix} .2 \\ .8 \end{bmatrix} \quad P(B|\neg a, d) = \begin{bmatrix} .3 \\ .7 \end{bmatrix}$$

$$P(B|a, \neg d) = \begin{bmatrix} .4 \\ .6 \end{bmatrix} \quad P(B|\neg a, \neg d) = \begin{bmatrix} .5 \\ .5 \end{bmatrix}.$$

More generally, constraints for the specifications of conditional mass functions relative to different nodes can be similarly represented by decision nodes which are the parents of these nodes (see for example Figure 4.4).

4.1.3 Decision-Theoretic Specification of Separately Specified Credal Networks

Finally, to provide a decision-theoretic specification, as required by Definition 4, of a separately specified credal network, it would suffice to reformulate the separately specified credal network as an extensive credal network whose tables are obtained considering all the combinations of the vertices of the separately specified conditional credal sets of the same variable.

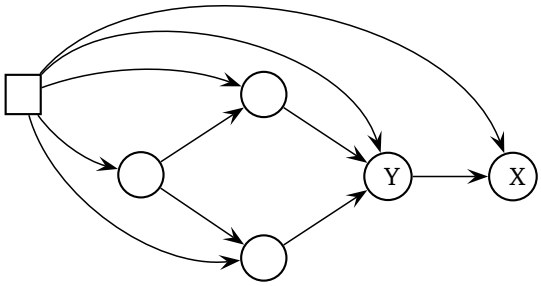


Figure 4.2: Decision-theoretic specification of a non-separately specified credal network over the DAG in Figure 4.1. Remember that circles denote uncertain nodes, while the square is used for the decision node.

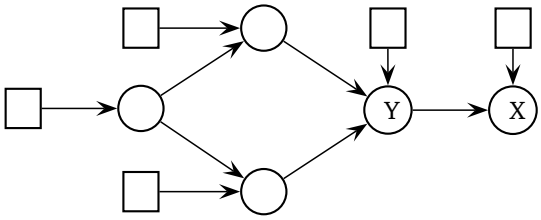


Figure 4.3: Decision-theoretic specification of an extensive credal network over the DAG in Figure 4.1.

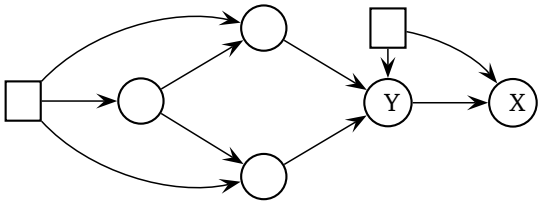


Figure 4.4: Decision-theoretic specification of a non-separately specified credal network over the DAG in Figure 4.1. Constraints between the specifications of the conditional credal sets of the nodes X and Y, and also between the three remaining nodes are assumed.

As an example, let us consider the node X of a separately specified credal network defined over the DAG in Figure 4.1. Assume Y to be binary and both the credal sets $K(X|y)$ and $K(X|\neg y)$ to be made of three extreme mass functions. A requirement for the probability table $P(X|Y)$ to belong to a set of nine tables, obtained from all the possible combinations where the first column takes values in $\text{ext}[K(X|y)]$ and the second in $\text{ext}[K(X|\neg y)]$, is clearly equivalent to leave the conditional probability mass functions $P(X|y)$ and $P(X|\neg y)$ to vary in the relative credal sets.

Yet, this approach suffers for an obvious exponential explosion (of the number of tables) in the input size. A more effective procedure consists in adding a decision node in between each (uncertain) node and its parents, according to the following graphical transformation.

Transformation 2. *Obtain a DAG $\mathcal{G}'$ from a DAG $\mathcal{G}$ over* **X** *by iterating, for each $X_i \in$* **X***, the following operations: (i) add a decision node D_i; (ii) draw an arc from each parent of X_i to D_i; (iii) delete the arcs connecting the parents of X_i with X_i; (iv) draw an arc from D_i to X_i.*

It is straightforward to check that Transformation 2 requires only a number of operations linear in the input size.

Given a separately specified credal network $\langle \mathcal{G}, \mathbb{K} \rangle$ over **X**, it is possible to consider a decision-theoretic specification of a credal network $\langle \mathcal{G}', \mathbb{O}, \mathbb{P}' \rangle$, where $\mathcal{G}'$ is the DAG returned by Transformation 2, **D** is the set of decision nodes (one for each node) added by the same transformation. To complete the decision-theoretic specification proceed as follows. For each uncertain node X_i, consider the set $\bigcup_{\pi_i \in \Omega_{\Pi_i}} \text{ext}[K(X_i|\pi_i)]$, i.e., the union of the extreme mass functions of *all* the conditional credal sets specified for X_i. Let the states $d_i \in \Omega_{D_i}$ of the corresponding decision node D_i index the elements of this set. Accordingly, for each uncertain node X_i, the conditional mass function $P'(X_i|d_i)$ corresponds to the vertex of the conditional credal set $K(X_i|\pi_i)$ associated to d_i, for each $d_i \in \Omega_{D_i}$. Regarding decision nodes, for each decision node D_i and the related value π_i of the parents, we simply set the subset $\Omega_{D_i}^{\pi_i} \subseteq \Omega_{D_i}$ to be such that $\{P'(X_i|d_i)\}_{d_i \in \Omega_{D_i}^{\pi_i}}$ are the vertices of $K(X_i|\pi_i)$. For this approach, which is clearly polynomial in the overall number of vertices of the conditional credal sets in $\mathbb{K}$, we have taken inspiration from *probability trees* representations, as defined in [CM02].[2]

As an example, consider the decision-theoretic specification of a separately specified credal network over the DAG $\mathcal{G}$ in Figure 4.1. The output $\mathcal{G}'$ returned

[2]It should be pointed out that the probability tree representation is different as it adds a variable for each configuration of the parents. Nevertheless the complexity of the representation does not increase as probability trees can represent asymmetrical irrelevance relationships.

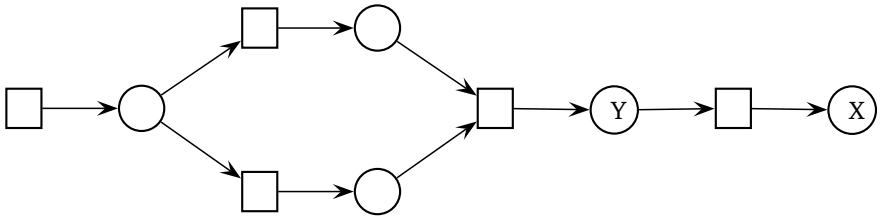


Figure 4.5: Decision-theoretic specification of a separately specified credal network over the DAG in Figure 4.1.

by Transformation 2 is the DAG in Figure 4.5. Regarding the quantification, consider for instance the procedure for the node X. Assume Y to be binary, and let

$$\text{ext}[K(X|y)] = \{P_1(X|y), P_2(X|y), P_3(X|y)\}$$

and

$$\text{ext}[K(X|\neg y)] = \{P_4(X|\neg y), P_5(X|\neg y), P_6(X|\neg y)\}.$$

For the decision node D added in between X and Y, we set $\Omega_D := \{1,2,3,4,5,6\}$, and, for the subsets of Ω_D corresponding to the possible values of Y, $\Omega_D^y := \{1,2,3\}$ and $\Omega_D^{\neg y} := \{4,5,6\}$, whereas, regarding X, we set $P'(X|D = d) := P_d(X|y_d)$ for each $d \in \Omega_D$ (where clearly $y_d := y$ if $d \in \Omega_D^y$ and $y_d := \neg y$ if $d \in \Omega_D^{\neg y}$).

4.2 From Decision-Theoretic to Separate Specification of Credal Networks

The transformations described in Section 4.1.2 and in Section 4.1.3 can be used to obtain in polynomial time a decision-theoretic specification of a credal network of any kind.[3] In this section, we prove that any decision-theoretic specification of a credal network over $\mathbf{X}$ can equivalently be regarded as a separate specification of a credal network over $\mathbf{X}' := (\mathbf{X}_D, \mathbf{X})$. This transformation is technically straightforward: it is based on representing decision nodes by uncertain nodes (Figure 4.6) with vacuous conditional credal sets, as formalized below.

[3] As a side note, it is important to be aware that a credal set can have a very large number of vertices, and this can still be a source of computational problems for algorithms (such as those based on the CCM transformation) that explicitly enumerate the vertices of a net's credal sets. This is a well-know issue, which in the present setup is related to the possibly large number of states for the decision nodes in the decision-theoretic representation of a credal net.

Transformation 3. *Given a decision-theoretic specification of a credal network* $\langle \mathcal{G}', \mathbb{O}, \mathbb{P}' \rangle$ *over* $\mathbf{X}$*, obtain a separately specified credal network* $\langle \mathcal{G}', \mathbb{K} \rangle$ *over* $\mathbf{X}' := (\mathbf{X}_D, \mathbf{X})$*, where the conditional credal sets in* $\mathbb{K}$ *are as follows, for each* $X_i \in \mathbf{X}$ *and* $\pi_i \in \Omega_{\Pi_i}$*:*

$$K(X_i|\pi_i) := \begin{cases} P'(X_i|\pi_i) & \text{if } X_i \in \mathbf{X} \\ K_{\Omega_{X_i}^{\pi_i}}(X_i) & \text{if } X_i \in \mathbf{X}_D, \end{cases} \tag{4.7}$$

where $P'(X_i|\pi_i)$ *is the mass function specified in* $\mathbb{P}'$ *and* $K_{\Omega_{X_i}^{\pi_i}}(X_i)$ *the* vacuous *credal set for* $\Omega_{X_i}^{\pi_i}$*.*

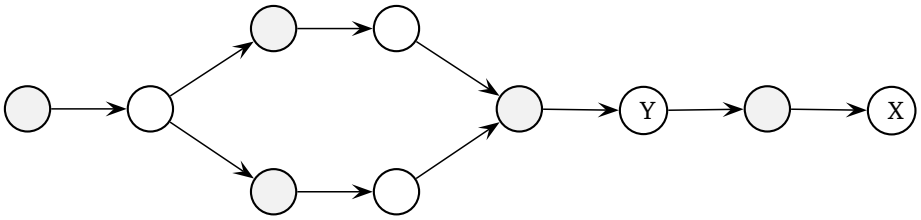


Figure 4.6: The DAG associated to the separately specified credal network returned by Transformation 3, from the decision-theoretic specification of the credal network based on the DAG in Figure 4.5. The conditional credal sets of the white nodes (corresponding to the original uncertain nodes) are precisely specified, while the gray nodes (i.e., new uncertain nodes corresponding to the former decision nodes) represent variables whose conditional credal sets are vacuous.

The (strong) relation between a decision-theoretic specification of a credal network $\langle \mathcal{G}', \mathbb{O}, \mathbb{P}' \rangle$ over $\mathbf{X}$ and the separately specified credal network $\langle \mathcal{G}', \mathbb{K} \rangle$ over $\mathbf{X}' := (\mathbf{X}_D, \mathbf{X})$ returned by Transformation 3 is outlined by the following result:

Theorem 4. *Let* $\tilde{K}(\mathbf{X})$ *be the marginal for* $\mathbf{X}$ *of the strong extension* $\tilde{K}(\mathbf{X}')$ *of* $\langle \mathcal{G}', \mathbb{K} \rangle$ *and* $K(\mathbf{X})$ *the strong extension of* $\langle \mathcal{G}', \mathbb{O}, \mathbb{P}' \rangle$*. Then:*

$$K(\mathbf{X}) = \tilde{K}(\mathbf{X}). \tag{4.8}$$

Proof. Consider a vertex of the strong extension of $\langle \mathcal{G}', \mathbb{K} \rangle$, i.e., a joint mass function $\tilde{P}(\mathbf{X}') \in \text{ext}[\tilde{K}(\mathbf{X}')]$. According to Proposition 1, $\tilde{P}(\mathbf{X}')$ can be obtained by the product, as in Equation (2.8), of a combination of vertices of the conditional credal sets in $\mathbb{K}$. Thus, for each $X_i \in \mathbf{X}_D$ and $\pi_i \in \Omega_{\Pi_i}$, $\tilde{P}(X_i|\pi_i)$ is a vertex

of the vacuous credal set $K_{\Omega_{X_i}^{\pi_i}}(X_i)$, i.e., a degenerate mass function over X_i assigning all the mass to a single $\tilde{x}_i \in \Omega_{X_i}$. Consider, on $\langle \mathcal{G}', \mathbb{O}, \mathbb{P}' \rangle$, the decision function f_{X_i} of X_i such that $f_{X_i}(\pi_i) = \tilde{x}_i$ for each $\pi_i \in \Omega_{\Pi_i}$. Let $\tilde{\mathbf{s}} \in \Omega_{\mathbf{S}}$ be the strategy corresponding to the array of decision functions selected in this way for each $X_i \in \mathbf{X}_D$. Clearly $P_{\tilde{\mathbf{s}}}(\mathbf{X}') = \tilde{P}(\mathbf{X}')$. Thus, considering all the vertices of $\tilde{K}(\mathbf{X}')$, we conclude $\tilde{K}(\mathbf{X}') \subseteq K(\mathbf{X}')$.

In order to prove the inverse inclusion, given a strategy $\mathbf{s} \in \Omega_{\mathbf{S}}$, consider the joint mass function $P_{\mathbf{s}}(\mathbf{X}')$ associated to the Bayesian network $\langle \mathcal{G}', \mathbb{P}'_{\mathbf{s}} \rangle$. As shown in Section 4.1.1, the elements of $\mathbb{P}'_{\mathbf{s}}$ corresponding to the nodes $X_i \in \mathbf{X}$ are just the same conditional probability mass functions $P'(X_i|\pi_i)$ specified in Equation (4.7). On the other side, the elements of $\mathbb{P}'_{\mathbf{s}}$ corresponding to the nodes $X_i \in \mathbf{X}_D$ are (degenerate) mass functions reproducing the decision functions of $\mathbf{s}$, and should therefore belong to the vacuous credal sets in Equation (4.7). Thus, $P_{\mathbf{s}}(\mathbf{X}') \in \tilde{K}(\mathbf{X}')$, and hence $K(\mathbf{X}') \subseteq \tilde{K}(\mathbf{X}')$.

Overall we have $K(\mathbf{X}') = \tilde{K}(\mathbf{X}')$, from which the thesis follows by a simple marginalization. □

From Theorem 4, it is straightforward to obtain the following result:

Corollary 2. *Any inference problem on a credal network obtained by a decision-theoretic specification can be equivalently solved in the separately specified credal network returned by Transformation 3.*

Let us stress that Transformation 3 is very simple, and it is surprising that it is presented here for the first time, as it is really the key to "separate" the credal sets of non-separately specified nets: in fact, given a non-separately specified credal network, one can obtain a decision-theoretic specification using the prescriptions of Section 4.1.2, and apply Transformation 3 to obtain a separately specified credal network. According to Corollary 2, then, any inference problem on the original credal network can equivalently be represented on this new separately specified credal network. In the following sections, two examples of applications of this procedure are presented. [4]

4.3 An Application: 2U for Extensive Specifications

The NP-hardness of credal networks belief updating has been proved even for singly connected topologies [dCC05]. Nevertheless, a singly connected credal

[4] It should be pointed out that the transformation described in Section 4.1.3, returning a decision-theoretic specification of a separately specified credal network, is not the inverse of Transformation 3, as the sequential application of the two transformations produces a model defined over a larger domain.

network with binary variables can be efficiently updated by the *2U algorithm* [FZ98]. At the present moment, 2U is the only polynomial-time algorithm for exact updating of credal networks, but it is designed only for separately specified credal networks. Here we show how 2U can be readily extended to deal exactly and efficiently also with extensively specified credal networks.

Consider a singly connected credal network as in Figure 4.7.a, defined over a set of binary variables with extensive specification of the conditional probability tables. According to the discussion of Section 4.1.2, a decision-theoretic specification of this credal network can be obtained by simply adding to each node a decision parent indexing the different tables. Instead of a single decision parent, a set of binary decision parents whose joint states correspond to the tables can be equivalently adopted (Figure 4.7.b). This means that, for example, a set of four probability tables providing an extensive specification of a node can be indexed by a single decision parent with four states, or by the joint states of two binary decision parents. Clearly, if the number of tables is not an integer power of two, this procedure introduces a number of redundant joint states for the decision parents.

Finally, from the decision-theoretic specification, we obtain a separately specified credal network through Transformation 3 (Figure 4.7.c). The overall procedure preserves the topology of the credal network, which remains singly connected, and is still defined over binary variables only. We can therefore update the credal network by 2U without making any change to the algorithm itself.

4.4 Application to Conservative Inference Rule

As a more involved application of the results in Sections 4.1 and Section 4.2, let us consider the CIR-based updating problem detailed in Section 3.1. In Section 4.4.1, the problem is mapped into a standard updating problem on a separately specified credal network, which can be updated by standard techniques. The result follows from a general equivalence relation regarding CIR-based inference in general. In the special case of updating, we also obtain, by similar transformations, a hardness proof in Section 4.4.2.

4.4.1 Algorithms for CIR-Based Inference

Consider a Bayesian network over $\mathbf{X} := (\tilde{\mathbf{X}}, \mathbf{X}_I)$, assigning positive probability to any joint state, where $\mathbf{X}_I$ are the variables missing by a process we do not know. As shown in Equation (3.1) in the special case of updating, CIR-based inference requires the evaluation of all the possible completions of the missing variables

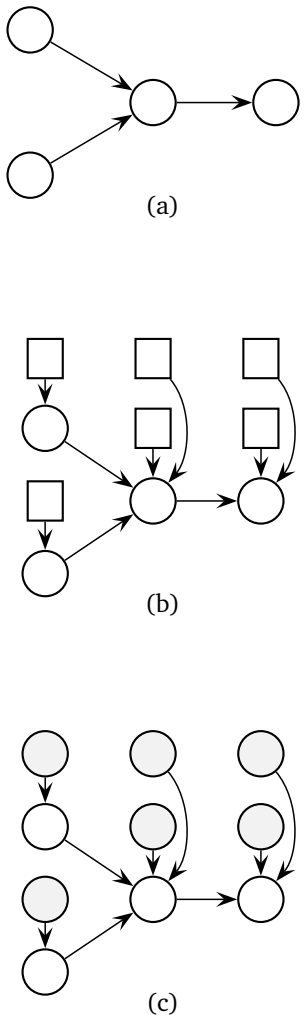


Figure 4.7: (a) A singly connected credal network over four binary variables; (b) its decision-theoretic specification with binary decision parents, assuming extensive specifications by sets of two tables for the root nodes, and four tables for the others; (c) the separately specified credal network returned by Transformation 3.

$\mathbf{X}_I$. This is equivalent to making inferences using the following credal set:

$$K_{\mathbf{X}_I}(\tilde{\mathbf{X}}) := \mathrm{CH}\{P(\tilde{\mathbf{X}}|\mathbf{x}_I)\}_{\mathbf{x}_I \in \Omega_{\mathbf{X}_I}}, \tag{4.9}$$

where the conditional mass function $P(\tilde{\mathbf{X}}|\mathbf{x}_I)$ is obtained from the joint mass function $P(\mathbf{X})$ associated to the Bayesian network.

The Bayesian network becomes an extensively specified credal network over $(\tilde{\mathbf{X}}, \mathbf{X}_I, \mathbf{X}'_I)$ after the transformation B2C defined in Section 3.1.2. A decision-theoretic specification of this non-separately specified credal network can be indeed obtained by simply adding to each node $X' \in \mathbf{X}'_I$ a decision parent of X', say X'', indexing the different tables. Such decision-theoretic credal network specification corresponding to the CIR-based inference problem is considered in the following theorem.

Theorem 5. *The following equivalence between credal sets holds:*

$$K_{\mathbf{X}_I}(\tilde{\mathbf{X}}) = K(\tilde{\mathbf{X}}|\overline{\mathbf{x}}'_I), \tag{4.10}$$

where the conditional credal set on the right-hand side is obtained from the strong extension of the decision-theoretic credal network specification corresponding to the CIR-based inference problem.

Proof. Let $P(\tilde{\mathbf{X}}, \mathbf{X}_I)$ be the joint probability mass function associated to the Bayesian network. Let also $\mathbf{X}''_I$ denote the array of decision nodes of the credal network. For each $X \in \mathbf{X}_I$, the corresponding elements of $\mathbf{X}'_I$ and $\mathbf{X}''_I$ are indicated as X' and X'', i.e., X' is the auxiliary child added to X according to the transformation B2C defined in Section 3.1.2,and X'' is the decision parent added to X' according to the prescriptions in Section 4.1.2. Let also $\overline{x}' \in \Omega_{X'}$ denote the component corresponding to X' of $\overline{\mathbf{x}}'_I \in \Omega_{\mathbf{X}'_I}$ (i.e., the value of the binary variable X' corresponding to the first row of the tables $P(X'|X)$ in Equation (3.2)). Note that X'' indexes the set of probability tables $P(X'|X)$ in Equation (3.2), which are in correspondence with the elements of Ω_X. We can therefore set $\Omega_{X''} := \Omega_X$ and regard the state $X'' = x$, for each $x \in \Omega_X$, as the index of the table in Equation (3.2) such that $P(X' = \overline{x}'|X = x) = 1$. Thus, the elements of $\Omega_{\mathbf{X}''_I}$, indexing the compatible Bayesian networks of the credal network, can be identified with those of $\Omega_{\mathbf{X}_I}$. Let $P_{\mathbf{X}''_I = \hat{\mathbf{x}}_I}(\tilde{\mathbf{X}}, \mathbf{X}_I, \mathbf{X}'_I)$ denote, for each $\hat{\mathbf{x}}_I \in \Omega_{\mathbf{X}''_I}$, the joint probability mass function of the compatible Bayesian network corresponding to $\hat{\mathbf{x}}_I$. The following factorization clearly holds:

$$P_{\mathbf{X}''_I = \hat{\mathbf{x}}_I}(\tilde{\mathbf{x}}, \mathbf{x}_I, \mathbf{x}'_I) = P(\tilde{\mathbf{x}}, \mathbf{x}_I) \cdot \prod_{X \in \mathbf{X}_I} \left[P_{\mathbf{X}''_I = \hat{\mathbf{x}}_I}(x'|x) \right]. \tag{4.11}$$

According to Equation (3.2), we have:

$$P_{\hat{\mathbf{X}}''_I = \hat{\mathbf{x}}_I}(\overline{x}'|x) = \delta_{x,\hat{x}}, \tag{4.12}$$

where $\delta_{x,\hat{x}}$ is equal to one if and only if $x = \hat{x}$ and zero otherwise. Thus, from Equation (4.11) and Equation (4.12), it follows that:

$$\sum_{\mathbf{x}_I \in \Omega_{\mathbf{X}_I}} P_{\mathbf{X}''_I = \hat{\mathbf{x}}_I}(\tilde{\mathbf{x}}, \mathbf{x}_I, \overline{\mathbf{x}}'_I) = P(\tilde{\mathbf{x}}, \hat{\mathbf{x}}_I). \tag{4.13}$$

From this we obtain

$$P_{\mathbf{X}''_I = \hat{\mathbf{x}}_I}(\tilde{\mathbf{X}} | \overline{\mathbf{x}}'_I) = P(\tilde{\mathbf{X}} | \mathbf{X}_I = \hat{\mathbf{x}}_I). \tag{4.14}$$

The thesis follows by simply considering Equation (4.14) for each $\hat{\mathbf{x}}_I \in \Omega_{\mathbf{X}_I}$. □

Equation (4.10) can be used to map general CIR-based inference problems in Bayesian networks into corresponding standard inferences on credal networks. The equivalence with respect to CIR-based updating in Equation (3.1) can be regarded as an obvious corollary of Theorem 5.

Finally, according to Theorem 4, the conditional credal set on the right-hand side of Equation (4.10) can be equivalently obtained from the strong extension of the separately specified credal network returned by Transformation 3. Overall, this procedure, which is illustrated in Figure 4.8, maps CIR-based inference problems in Bayesian networks into corresponding problems in separately specified credal networks, for which existing algorithms can be employed. An analogous procedure could be developed to address CIR-based inference problems on credal networks.

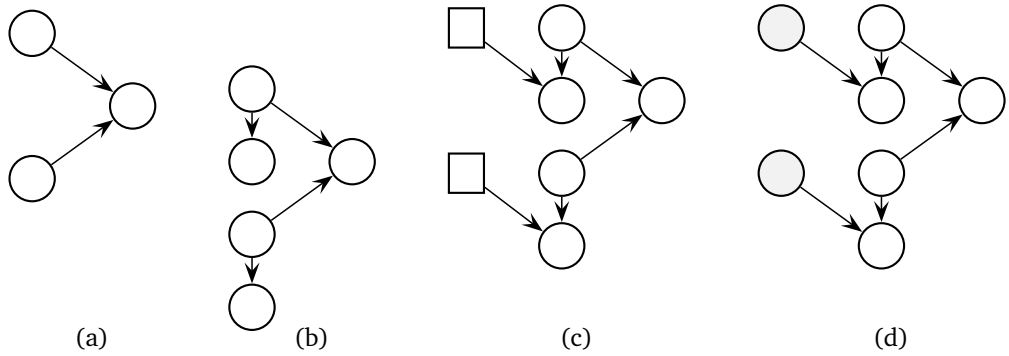


Figure 4.8: (a) A CIR-based inference problem on a Bayesian network where the missing variables $\mathbf{X}_I$ correspond to the root nodes; (b) the corresponding extensive credal network returned by the transformation B2C defined in Section 3.1.2; (c) the decision-theoretic specification of this extensive credal network; (d) the separately specified credal network returned by Transformation 3.

4.4.2 Hardness of CIR-Based Updating

In this section we determine the computational complexity of CIR-based updating on Bayesian networks.

To this end, consider the class of separately specified credal networks such that the specification of the non-root nodes is precise, i.e., the corresponding conditional credal sets are reduced to a single conditional probability mass function. These credal networks are called to be *with precise non-root nodes* and are considered in the following theorem.

Theorem 6. *Any updating problem on a credal network with precise non-root nodes can be mapped into a CIR-based updating problem on a Bayesian network in linear time.*

Proof. Consider a (separately specified) credal network over $\mathbf{X}$ with precise non-root nodes. A decision-theoretic specification of this credal network can be obtained by simply adding to each root node a decision parent node indexing the vertices of the unconditional credal set associated to this node. Let $\mathbf{X}_D$ be the decision nodes added to the credal network by the transformation. For each $\mathbf{x}_D \in \Omega_{\mathbf{X}_D}$, let $P_{\mathbf{x}_D}(\mathbf{X})$ be the joint probability mass function associated to the compatible Bayesian network indexed by $\mathbf{x}_D$.

Obtain a Bayesian network from the decision-theoretic specification of the credal network, by simply regarding the decision root nodes $\mathbf{X}_D$ as uncertain nodes for which uniform unconditional mass functions have been specified. Let $\tilde{P}(\mathbf{X}, \mathbf{X}_D)$ be the joint probability mass function associated to this Bayesian network. It is straightforward to check that, for each $\mathbf{x}_D \in \Omega_{X_D}$,

$$\tilde{P}(\mathbf{x}|\mathbf{x}_D) = P_{\mathbf{x}_D}(\mathbf{x}). \tag{4.15}$$

Thus, a generic updating problem on the credal network can be mapped into a CIR-based updating problem on this Bayesian network, by simply assuming the variables $\mathbf{X}_D$ to be missing by an unknown mechanism. □

Remarkably, the hardness proof of updating with credal networks reported in [dCC05, Theorem 3] is based on the reduction of a *Boolean satisfiability* [GJ79] problem to the updating of a (singly-connected) credal network with precise non-root nodes. According to Theorem 6, it is therefore straightforward to conclude the following result:

Corollary 3. *CIR-based updating on Bayesian networks is NP-hard.*

4.5 Summary and Conclusions

We have defined a new graphical language to formulate any type of credal network, both separately and non-separately specified. We have also shown that any net represented with the new language can be easily transformed into an equivalent separately specified credal net. This implies, in particular, that non-separately specified nets have an equivalent separately specified representation, for which updating algorithms are available in the literature.

Two examples of applications of this procedure have been detailed: the generalization of the 2U algorithm to the extensive case, and a general algorithmic procedure to solve CIR-based inference on Bayesian networks. Additionally, we have also exploited our formalism to prove the NP-hardness of CIR-based updating on Bayesian networks.

With respect to future work, many other developments seem to be possible. First of all it is important to note that the proposed transformation also shows that a subclass of separately specified credal networks can be used to solve inference problems for arbitrary specified credal nets: this is the class of nets in which the credal sets are either vacuous or precise. An important development of the approximate L2U algorithm is particularly suited just for such a class, and will be considered in Chapter 5. Finally, the strong connection between the language for credal networks introduced in this paper and the formalism of decision networks (including influence diagrams), seems to be particularly worth exploring for cross-fertilization between the two fields.

Chapter 5

Generalized Loopy 2U: A New Algorithm for Approximate Inference in Credal Networks

Credal networks generalize Bayesian networks relaxing numerical parameters. As described in Section 2.5, this considerably expands expressivity, but makes belief updating a hard task even on polytrees. Nevertheless, if all the variables are binary, polytree-shaped credal networks can be efficiently updated by the 2U algorithm (see Section 2.6.1). In the first part of this chapter we present a *binarization algorithm* that makes it possible to approximate an updating problem in a credal net by a corresponding problem in a credal net over binary variables. The procedure leads to outer bounds for the original problem. The binarized nets are in general multiply connected, but can be updated by the *loopy* variant of 2U. The overall procedure is very fast and provides relatively accurate inferences.

A significant improvement of the quality of this approximation is obtained in the second part of this chapter, where we develop a new efficient algorithm for approximate belief updating in credal nets. The algorithm is based on an important representation result we prove for general credal nets: that any credal net can be equivalently reformulated as a credal net with binary variables; moreover, the transformation, which is considerably more complex than in the Bayesian case, can be implemented in polynomial time. The equivalent binary credal net is then updated by L2U. Thus, we generalize L2U to non-binary credal nets, obtaining an accurate and scalable algorithm for the general case, called GL2U, which is approximate only because of its loopy nature. The accuracy of the inferences of the algorithm is evaluated by promising empirical tests.

5.1 Binarization Algorithms

As noted in Section 2.6.1, 2U is the only efficient algorithm for exact updating of credal networks. 2U has two main limitations: the topology of the network, which is assumed to be singly connected, and the number of possible states for the variables, which is limited to two for any variable. The limitation about topology is partially overcome by L2U, which can be employed to update multiply connected credal networks. Here, we overcome also the limitation of 2U about the number of possible states. To this end, a map is defined to transform a generic updating problem on a credal net into a second updating problem on a corresponding binary credal net. First, we show how to represent a random variable as a collection of binary variables (Section 5.1.1). Secondly, we employ this idea to represent a Bayesian network as an equivalent binary Bayesian network (Section 5.1.3) with an appropriate graphical structure (Section 5.1.2). Finally, we extend this binarization procedure to the case of credal networks (Section 5.1.4).[1]

5.1.1 Binarization of Variables

Assume d_i, which is the number of states for X_i, to be an integer power of two, i.e., $\Omega_{X_i} = \{x_{i0}, \ldots, x_{i(d_i-1)}\}$, with $d_i = 2^{m_i}$ and m_i integer. An obvious one-to-one correspondence between the states of X_i and the joint states of an array of m_i binary variables $(\tilde{X}_{i(m_i-1)}, \ldots, \tilde{X}_{i1}, \tilde{X}_{i0})$ can be established: we assume that the joint state $(\tilde{x}_{i(m_i-1)}, \ldots, \tilde{x}_{i0}) \in \{0,1\}^{m_i}$ is associated to $x_{il} \in \Omega_{X_i}$, where l is the integer whose m_i-bit binary representation is the sequence $\tilde{x}_{i(m_i-1)} \cdots \tilde{x}_1 \tilde{x}_0$. We refer to this procedure as the *binarization* of X_i and the binary variable $\tilde{X}_{ij}$ is called the *j-th order bit* of X_i. As an example, the state x_{i6} of X_i, assuming for X_i eight possible values, i.e., $m_i = 3$, would be represented by the joint state $(1,1,0)$ for the three binary variables $(\tilde{X}_{i2}, \tilde{X}_{i1}, \tilde{X}_{i0})$.

If the number of states of X_i is not an integer power of two, the variable is called *not binarizable*. In this case we can make X_i binarizable simply adding to Ω_{X_i} a number of *impossible states*[2] up the the nearest power of two. For example we can make binarizable a variable with six possible values by adding two impossible states. Clearly, once the variables of $\mathbf{X}$ have been made binarizable, there is an obvious one-to-one correspondence between the joint states of $\mathbf{X}$ and those of the array of the binary variables returned by the binarization of $\mathbf{X}$, say $\tilde{\mathbf{X}} = (\tilde{X}_{1(m_1-1)}, \ldots, \tilde{X}_{10}, \tilde{X}_{2(m_2-1)}, \ldots, \tilde{X}_{n(m_n-1)}, \ldots, \tilde{X}_{n0})$. Regarding notation,

[1] The work presented in this section has been done in cooperation with Jaime Shinsuke Ide and Fabio Gagliardi Cozman.

[2] This denomination is justified by the fact that, in the following, we will set the probabilities for these states equal to zero.

for each $\mathbf{x} \in \Omega_{\mathbf{X}}$, $\tilde{\mathbf{x}}$ is assumed to denote the corresponding element of $\Omega_{\tilde{\mathbf{X}}}$ and *vice versa*. Similarly, $\tilde{x}_E$ denotes the joint state for the bits of the nodes in X_E corresponding to x_E.

5.1.2 Graph Binarization

Let $\mathcal{G}$ be a DAG associated to a set of binarizable variables $\mathbf{X}$. We call the *binarization* of $\mathcal{G}$ with respect to $\mathbf{X}$, a second DAG $\tilde{\mathcal{G}}$ associated to the variables $\tilde{\mathbf{X}}$ returned by the binarization of $\mathbf{X}$, obtained with the following prescriptions: (i) two nodes of $\tilde{\mathcal{G}}$ corresponding to bits of different variables in $\mathbf{X}$ are connected by an arc if and only if there is an arc with the same orientation between the relative variables in $\mathbf{X}$; (ii) an arc connects two nodes of $\tilde{\mathcal{G}}$ corresponding to bits of the same variable of $\mathbf{X}$ if and only if the order of the bit associated to the node from which the arc departs is lower than the order of the bit associated to the remaining node.

Figure 5.1 reports a multiply connected DAG $\mathcal{G}$ and its binarization $\tilde{\mathcal{G}}$. As an example of Prescription (i) for $\tilde{\mathcal{G}}$, note the arcs connecting all the three bits of X_0 with all the two bits of X_2, while, considering the bits of X_0, the arcs between the bit of order zero and those of order one and two, as well as that between the bit of order one and that of order two, are drawn because of Prescription (ii).

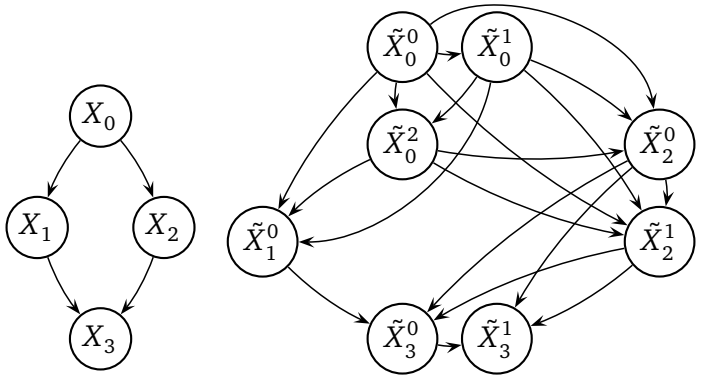


Figure 5.1: A multiply connected DAG (left) and its binarization (right) assuming $d_0 = 8$, $d_1 = 2$ and $d_2 = d_3 = 4$.

5.1.3 Bayesian Networks Binarization

The notion of "binarizability" extends to Bayesian networks as follows: $\langle \mathcal{G}, \mathbb{P} \rangle$ is *binarizable* if and only if $\mathbf{X}$ is a set of binarizable variables. A non-binarizable

Bayesian network can be made binarizable by the following procedure: (i) make the variables in $\mathbf{X}$ binarizable; (ii) specify zero values for the conditional probabilities of the impossible states, i.e., $P(x_{ij}|\pi_i) = 0$ for each $j \geq d_i$, for each $\pi_i \in \Omega_{\pi_i}$ and for each $i = 1,\ldots,n$; (iii) arbitrarily specify the mass function $P(X_i|\pi_i)$ for each π_i such that at least one of the states of the parents Π_i corresponding to π_i is an impossible state, for $i = 1,\ldots,n$. Considering Equation (2.1) and Prescription (ii), it is easy to note that, if the joint state $\mathbf{x} = (x_1,\ldots,x_n)$ of $\mathbf{X}$ is such that at least one of the states x_i, with $i = 1,\ldots,n$, is an impossible state, then $P(\mathbf{x}) = 0$, irrespectively of the values of the mass functions specified as in Prescription (iii). Thus, given a non-binarizable Bayesian network, the procedure described in this paragraph returns a binarizable Bayesian network that preserves the original probabilities. This makes possible to focus on the case of binarizable Bayesian networks without loss of generality, as in the following:

Definition 5. *Let $\langle \mathcal{G}, \mathbb{P} \rangle$ be a binarizable Bayesian network over $\mathbf{X}$. The binarization of $\langle \mathcal{G}, \mathbb{P} \rangle$ is a binary Bayesian network $\langle \tilde{\mathcal{G}}, \tilde{\mathbb{P}} \rangle$ over $\tilde{\mathbf{X}}$ obtained as follows: (i) $\tilde{\mathcal{G}}$ is the binarization of $\mathcal{G}$ with respect to $\mathbf{X}$ (ii) the joint probability mass function $\tilde{P}(\tilde{\mathbf{X}})$ associated to $\langle \tilde{\mathcal{G}}, \tilde{\mathbb{P}} \rangle$ corresponds to the following specifications of the conditional probabilities for the variables in $\tilde{\mathbf{X}}$ given their parents:*[3]

$$\tilde{P}(\tilde{x}_{ij}|\tilde{x}_{i(j-1)},\ldots,\tilde{x}_{i0},\tilde{\pi}_i) \propto \sum\nolimits_l^* P(x_{il}|\pi_i) \quad \begin{array}{l} i=1,\ldots,n \\ j=0,\ldots,m_i-1 \\ \pi_i \in \Omega_{\Pi_i}, \end{array} \tag{5.1}$$

where the sum $\sum^*$ is restricted to the states $x_{il} \in \Omega_{X_i}$ such that the first $j+1$ bits of the binary representation of l are $\tilde{x}_{i0},\ldots,\tilde{x}_{ij}$, π_i is the joint state of the parents of X_i corresponding to the joint state $\tilde{\pi}_i$ for the bits of the parents of X_i, and the symbol $\propto$ denotes proportionality.

The variables $(\tilde{X}_{i(j-1)},\ldots,\tilde{X}_{i0},\tilde{\Pi}_i)$ are clearly the parents of $\tilde{X}_{ij}$ according to $\tilde{\mathcal{G}}$. In the following, the joint state $(\tilde{x}_{i(j-1)},\ldots,\tilde{x}_{i0},\tilde{\pi}_i)$ will be denoted as $\pi_{\tilde{X}_{ij}}$.

As an example of the procedure described in Definition 5, let X_0 be a variable with four states associated to a parentless node of a Bayesian network. Assuming for the corresponding mass function

$$[P(x_{00}),P(x_{01}),P(x_{02}),P(x_{03})] = (.2,.3,.4,.1),$$

we can use Equation (5.1) to obtain the mass functions associated to the two bits of X_0 in the binarized Bayesian network. This leads to:

$$\tilde{P}(\tilde{X}_{00}) = (.6,.4),$$

[3] If the sum on the right-hand side of Equation (5.1) is zero for both the values of $\tilde{X}_{ij}$, the corresponding conditional mass function is arbitrary specified.

$$\tilde{P}(\tilde{X}_{01}|\tilde{X}_{00}=0)=(\frac{1}{3},\frac{2}{3}),$$
$$\tilde{P}(\tilde{X}_{01}|\tilde{X}_{00}=1)=(\frac{3}{4},\frac{1}{4}),$$

where an array notation $[P(\tilde{X}=0),P(\tilde{X}=1)]$ is employed to denote a mass function of a binary variable $\tilde{X}$.

A Bayesian network and its binarization are basically the same probabilistic model and we can represent any updated belief in the original Bayesian network as a corresponding belief in the binarized Bayesian network, according to the following:

Theorem 7. *Let $\langle \mathcal{G}, \mathbb{P}\rangle$ be a binarizable Bayesian network and $\langle \tilde{\mathcal{G}}, \tilde{\mathbb{P}}\rangle$ its binarization. Then, given a queried variable $X_q \in \mathbf{X}$ and an evidence $X_E = x_E$:*

$$P(x_q|x_E)=\tilde{P}(\tilde{x}_{q(m_q-1)}\dots\tilde{x}_{q0}|\tilde{x}_E), \tag{5.2}$$

where $(\tilde{x}_{q(m_q-1)},\dots,\tilde{x}_{q0})$ is the joint state of the bits of X_q corresponding to x_q.

Proof. With some algebra it is easy to check that the inverse of Equation (5.1) is:

$$P(x_{il}|\pi_i)=\prod_{j=0}^{m_i-1}\tilde{P}(\tilde{x}_{ij}|\tilde{x}_{i(j-1)},\dots,\tilde{x}_{i0},\tilde{\pi}_i), \tag{5.3}$$

where $(\tilde{x}_{i(m_i-1)},\dots,\tilde{x}_{i0})$ is the m_i-bit binary representation of l. Thus, $\forall \mathbf{x}\in\Omega_{\mathbf{X}}$:

$$P(\mathbf{x})=\prod_{i=1}^{n}P(x_i|\pi_i)=\prod_{i=1}^{n}\prod_{j=0}^{m_i-1}\tilde{P}(\tilde{x}_{ij}|\tilde{x}_{i(j-1)},\dots,\tilde{x}_{i0},\tilde{\pi}_i)=\tilde{P}(\tilde{\mathbf{x}}), \tag{5.4}$$

where the first passage is because of Equation (2.1), the second because of Equation (5.3) and the third because of the Markov condition for the binarized Bayesian network. Thus:

$$P(x_q|x_E)=\frac{P(x_q,x_E)}{P(x_E)}=\frac{\tilde{P}(\tilde{x}_{q(m_q-1)},\dots,\tilde{x}_{q0},\tilde{x}_E)}{\tilde{P}(\tilde{x}_E)}, \tag{5.5}$$

that proves the thesis as in Equation (5.2). □

5.1.4 Extension to Credal Networks

In order to generalize the binarization from Bayesian networks to non-separately specified credal networks, we first extend the notion of binarizability: a credal network $\langle \mathcal{G}, \mathbb{K}\rangle$ over $\mathbf{X}$ is called binarizable if and only if $\mathbf{X}$ is binarizable. A

non-binarizable credal network can be made binarizable by the following procedure: (i) make the variables in $\mathbf{X}$ binarizable; (ii) specify zero upper (and lower) probabilities for conditional probabilities of the impossible states: $\underline{P}(x_{ij}|\pi_i) = \overline{P}(x_{ij}|\pi_i) = 0$ for each $j \geq d_i$, for each $\pi_i \in \Omega_{\Pi_i}$, and for each $i = 1,\ldots,n$; (iii) arbitrarily specify the conditional credal sets $K(X_i|\pi_i)$ for each π_i such that at least one of the states of the parents Π_i corresponding to π_i is an impossible state, for $i = 1,\ldots,n$. According to Prescription (i), it is easy to check that, if the joint state $\mathbf{x} = (x_1,\ldots,x_n)$ of $\mathbf{X}$ is such that at least one of the states x_i, with $i = 1,\ldots,n$, is an impossible state, then $P(\mathbf{x}) = 0$, irrespectively of the conditional credal sets specified as in the Prescription (iii), and for each $P(\mathbf{X}) \in K(\mathbf{X})$. Thus, given a non-binarizable credal network, the procedure described in this paragraph returns a binarizable credal network, that preserves the original probabilities. This makes possible to focus on the case of binarizable credal networks without loss of generality, as in the following:

Definition 6. *Let $\langle \mathcal{G}, \mathbf{P}(\mathbf{X})\rangle$ be a binarizable credal network.*[4] *The binarization of $\langle \mathcal{G}, \mathbf{P}(\mathbf{X})\rangle$ is a binary credal network $\langle \tilde{\mathcal{G}}, \tilde{\mathbf{P}}(\tilde{\mathbf{X}})\rangle$, with $\tilde{\mathcal{G}}$ binarization of $\mathcal{G}$ with respect to $\mathbf{X}$ and the following separate specifications of the extreme probabilities:*[5]

$$\underline{\tilde{P}}(\tilde{x}_{ij}|\pi_{\tilde{X}_{ij}}) \equiv \min_{k=1,\ldots,m} \tilde{P}_k(\tilde{x}_{ij}|\pi_{\tilde{X}_{ij}}), \tag{5.6}$$

where $\langle \tilde{\mathcal{G}}, \tilde{P}_k(\tilde{\mathbf{X}})\rangle$ is the binarization of $\langle \mathcal{G}, P_k(\mathbf{X})\rangle$ for each $k = 1,\ldots,m$.

Definition 6 implicitly requires the binarization of all the Bayesian networks $\langle \mathcal{G}, P_k(\mathbf{X})\rangle$ associated to $\langle \mathcal{G}, \mathbf{P}(\mathbf{X})\rangle$, but the right-hand side of Equation (5.6) is not a minimum over all the Bayesian networks associated to a $\langle \tilde{\mathcal{G}}, \tilde{\mathbf{P}}(\tilde{\mathbf{X}})\rangle$, being in general $\tilde{P}(\tilde{\mathbf{X}}) \neq \{\tilde{P}_k(\tilde{\mathbf{X}})\}_{k=1}^m$. This means that it is not possible to represent an updating problem in a credal network as a corresponding updating problem in the binarization of the credal network, and we should therefore regard $\langle \tilde{\mathcal{G}}, \tilde{\mathbf{P}}(\tilde{\mathbf{X}})\rangle$ as an approximate description of $\langle \mathcal{G}, \mathbf{P}(\mathbf{X})\rangle$.

Remarkably, according to Equation (5.1), the conditional mass functions for the bits of X_i relative to the value $\tilde{\pi}_i$, can be obtained from the single mass function $P(X_i|\pi_i)$. Therefore, if we use Equation (5.1) with $P_k(\mathbf{X})$ in place of $P(\mathbf{X})$ for each $k = 1,\ldots,m$ to compute the probabilities $\tilde{P}_k(\tilde{x}_{ij}|\pi_{\tilde{X}_{ij}})$ in Equation (5.6), the

[4] In the following, we adopt the compact notation $\mathbf{P}(\mathbf{X})$ to denote the collection of joint probability mass functions corresponding to the compatible Bayesian networks of a credal network. Accordingly, we denote a credal network as $\langle \mathcal{G}, \mathbf{P}(\mathbf{X})\rangle$ and a Bayesian network as $\langle \mathcal{G}, P(\mathbf{X})\rangle$.

[5] Note that in the case of a binary variables a specification of the extreme probabilities as in Equation (5.6) is equivalent to the explicit specification of the (two) vertices of the conditional credal set $K(\tilde{X}_{ij}|\pi_{\tilde{X}_{ij}})$: if $\tilde{X}$ is a binary variable and we specify $\underline{P}(\tilde{X} = 0) = s$ and $\underline{P}(\tilde{X} = 1) = t$, then the credal set $K(\tilde{X})$ is the convex hull of the mass functions $P_1(\tilde{X}) = (s, 1-s)$ and $P_2(\tilde{X}) = (1-t, t)$.

only mass function required to do such calculations is $P_k(X_i|\pi_i)$. Thus, instead of considering all the joint mass functions $P_k(\mathbf{X})$, with $k = 1, \ldots, m$, we can restrict our attention to the conditional mass functions $P(X_i|\pi_i)$ associated to the elements of the conditional credal set $K(X_i|\pi_i)$ and take the minimum, i.e.,

$$\underline{\tilde{P}}(\tilde{x}_{ij}|\pi_{\tilde{X}_{ij}}) = \min_{P(X_i|\pi_i)\in K(X_i|\pi_i)} \tilde{P}(\tilde{x}_{ij}|\pi_{\tilde{X}_{ij}}), \tag{5.7}$$

where $\tilde{P}(\tilde{x}_{ij}|\pi_{\tilde{X}_{ij}})$ is obtained from $P(X_i|\pi_i)$ using Equation (5.1) and the minimization on the right-hand side of Equation (5.7) can be clearly restricted to the vertices of $K(X_i|\pi_i)$. The procedure is therefore linear in the input size.

As an example, let X_0 be a variable with four possible states associated to a parentless node of a credal network. Assuming that the credal set $K(X_0)$ is the convex hull of the mass functions $(.2, .3, .4, .1)$, $(.25, .25, .25, .25)$, and $(.4, .2, .3, .1)$, we can use Equation (5.1) to compute the mass functions associated to the two bits of X_0 for each vertex of $K(X_0)$ and then consider the minima as in Equation (5.7), obtaining: $\underline{\tilde{P}}(\tilde{X}_{00}) = (.5, .3)$, $\underline{\tilde{P}}(\tilde{X}_{01}|\tilde{X}_{00} = 0) = (\frac{1}{3}, \frac{3}{7})$, $\underline{\tilde{P}}(\tilde{X}_{01}|\tilde{X}_{00} = 1) = (\frac{1}{2}, \frac{1}{4})$.

The equivalence between an updating problem in a Bayesian network and in its binarization as stated by Theorem 7 is generalizable in an approximate way to the case of credal networks, as stated by the following:

Theorem 8. *Let $\langle \mathcal{G}, \mathbf{P}(\mathbf{X})\rangle$ be a binarizable credal network and $\langle \tilde{\mathcal{G}}, \tilde{\mathbf{P}}(\tilde{\mathbf{X}})\rangle$ its binarization. Then, given a queried variable $X_q \in \mathbf{X}$ and an evidence $X_E = x_E$:*

$$\underline{P}(x_q|x_E) \geq \underline{\tilde{P}}(\tilde{x}_{q(m_q-1)}, \ldots, \tilde{x}_{q0}|\tilde{x}_E), \tag{5.8}$$

where $(\tilde{x}_{q(m_q-1)}, \ldots, \tilde{x}_{q0})$ is the joint state of the bits of X_q corresponding to x_q.

In order to prove Theorem 8, we first need the following result.

Lemma 1. *Let $\{\langle \mathcal{G}, P_k(\mathbf{X})\rangle\}_{k=1}^m$ be the Bayesian networks associated to a credal network $\langle \mathcal{G}, \mathbf{P}(\mathbf{X})\rangle$. Let also $\langle \tilde{\mathcal{G}}, \tilde{\mathbf{P}}(\tilde{\mathbf{X}})\rangle$ be the binarization of $\langle \mathcal{G}, \mathbf{P}(\mathbf{X})\rangle$. Then, the Bayesian network $\langle \tilde{\mathcal{G}}, \tilde{P}_k(\tilde{\mathbf{X}})\rangle$, which is the binarization of $\langle \mathcal{G}, P_k(\mathbf{X})\rangle$, specifies a joint mass function that belongs to the strong extension of $\langle \tilde{\mathcal{G}}, \tilde{\mathbf{P}}(\tilde{\mathbf{X}})\rangle$, i.e.,*

$$\tilde{P}_k(\tilde{\mathbf{X}}) \in \tilde{K}(\tilde{\mathbf{X}}), \tag{5.9}$$

for each $k = 1, \ldots, m$, with $\tilde{K}(\tilde{\mathbf{X}})$ denoting the strong extension of $\langle \tilde{\mathcal{G}}, \tilde{\mathbf{P}}(\tilde{\mathbf{X}})\rangle$.

Proof. As noted in Section 2.4, the strong extension of $\langle \tilde{\mathcal{G}}, \tilde{\mathbf{P}}(\tilde{\mathbf{X}})\rangle$ is:

$$\tilde{K}(\tilde{\mathbf{X}}) := \mathrm{CH}\Big\{ \prod_{\tilde{X}_{ij}\in\tilde{\mathbf{X}}} \tilde{P}(\tilde{X}_{ij}|\Pi_{\tilde{X}_{ij}}) : \tilde{P}(\tilde{X}_{ij}|\pi_{\tilde{X}_{ij}}) \in \tilde{K}(\tilde{X}_{ij}|\pi_{\tilde{X}_{ij}}) \quad \begin{matrix} \forall \pi_{\tilde{X}_{ij}} \in \Omega_{\Pi_{\tilde{X}_{ij}}} \\ \forall \tilde{X}_{ij} \in \tilde{\mathbf{X}} \end{matrix} \Big\}. \tag{5.10}$$

On the other side, considering the Markov condition for $\langle \tilde{\mathcal{G}}, \tilde{P}_k(\tilde{\mathbf{X}})\rangle$, we have:

$$\tilde{P}_k(\mathbf{X}) = \prod_{\tilde{X}_{ij}\in\tilde{\mathbf{X}}} \tilde{P}_k(\tilde{X}_{ij}|\Pi_{\tilde{X}_{ij}}). \tag{5.11}$$

But, for each $\pi_{\tilde{X}_{ij}} \in \Omega_{\Pi_{ij}}$ and $\tilde{X}_{ij} \in \tilde{\mathbf{X}}$, the conditional mass function $\tilde{P}_k(\tilde{X}_{ij}|\pi_{\tilde{X}_{ij}})$ belongs to the conditional credal set $\tilde{K}(\tilde{X}_{ij}|\pi_{\tilde{X}_{ij}})$ because of Equation (5.6). Thus, the joint mass function in Equation (5.11) belongs to the set in Equation (5.10), and that holds for each $k = 1, \ldots, m$. □

Lemma 1 basically states an inclusion relation between the strong extension of $\langle \tilde{\mathcal{G}}, \tilde{\mathbf{P}}(\tilde{\mathbf{X}})\rangle$ and the set of joint mass functions $\{P_k(\mathbf{X})\}_{k=1}^m$, which, according to the equivalence in Equation (5.4), is just an equivalent representation of $\langle \mathcal{G}, \mathbf{P}(\mathbf{X})\rangle$. This will be used to prove the relation between inferences in a credal network and in its binarization as stated by Theorem 8.

Proof of Theorem 8. We have:

$$\underline{P}(x_q|x_E) = \min_{k=1,\ldots,m} P_k(x_q|x_E) = \min_{k=1,\ldots,m} \tilde{P}_k(\tilde{x}_{q(m_q-1)}, \ldots, \tilde{x}_{q0}|\tilde{x}_E), \tag{5.12}$$

where the first passage is because of Equations (2.10) and (2.2), and the second because of Theorem 7 referred to the Bayesian network $\langle \mathcal{G}, P_k(\mathbf{X})\rangle$, for each $k = 1, \ldots, m$.

On the other side, the lower posterior probability probability on the right-hand side of Equation (5.8) can be equivalently expressed as:

$$\underline{P}(\tilde{x}_{q(m_q-1)}, \ldots, \tilde{x}_{q0}|\tilde{x}_E) = \min_{\tilde{P}(\tilde{\mathbf{X}})\in\tilde{K}(\tilde{\mathbf{X}})} \tilde{P}(\tilde{x}_{q(m_q-1)}, \ldots, \tilde{x}_{q0}|\tilde{x}_E), \tag{5.13}$$

where $\tilde{K}(\tilde{\mathbf{X}})$ is the strong extension of $\langle \tilde{\mathcal{G}}, \tilde{\mathbf{P}}(\tilde{\mathbf{X}})\rangle$. Considering the minima on the right-hand sides of Equations (5.12) and (5.13), we observe that they refer to the same function and the first minimum is over a domain that is included in that of the second because of Lemma 1. Thus, the lower probability in Equation (5.12) cannot be less than that on Equation (5.13), that is the thesis. □

The inequality in Equation (5.8) together with its analogous for the upper probabilities provides an outer bound for the posterior interval associated to a generic updating problem in a credal network. Such approximation is the posterior interval for the corresponding problem on the binarized credal network.

Note that L2U cannot update joint states of two or more variables: this means that we can compute the right-hand side of Equation (5.8) by a direct application of L2U only in the case $m_q = 1$, i.e, if the queried variable X_q is binary.

If X_q has more than two possible states, a simple transformation of the binarized credal network is necessary to apply L2U. The idea is simply to define an additional binary random variable, which is true if and only if

$$(\tilde{X}_{q(m_q-1)},\ldots,\tilde{X}_{q0}) = (\tilde{x}_{q(m_q-1)},\ldots,\tilde{x}_{q0}).$$

This variable is a deterministic function of some of the variables in $\tilde{\mathbf{X}}$, and can therefore be easily embedded in the credal network $\langle \tilde{\mathcal{G}}, \tilde{\mathbf{P}}(\tilde{\mathbf{X}})\rangle$. We simply add to $\tilde{\mathcal{G}}$ a binary node, say $C_{\tilde{x}_{q(m_q-1)},\ldots,\tilde{x}_{q0}}$, with no children and whose parents are $\tilde{X}_{q(m_q-1)},\ldots,\tilde{X}_{q0}$, and specify the probabilities for the state 1 (true) of $C_{\tilde{x}_{q(m_q-1)},\ldots,\tilde{x}_{q0}}$, conditional on the values of its parents $\tilde{X}_{q(m_q-1)},\ldots,\tilde{X}_{q0}$, equal to one only for the joint value of the parents $(\tilde{x}_{q(m_q-1)},\ldots,\tilde{x}_{q0})$ and zero otherwise. Then, it is straightforward to check that:

$$\underline{\tilde{P}}(\tilde{x}_{q(m_q-1)},\ldots,\tilde{x}_{q0}|\tilde{x}_E) = \underline{\tilde{P}}'(C_{\tilde{x}_{q(m_q-1)},\ldots,\tilde{x}_{q0}} = 1|\tilde{x}_E), \tag{5.14}$$

where $\underline{P}'$ denotes the lower probability in the credal network with the additional node. Thus, according to Equation (5.14), if X_q has more than two possible values, we simply add the node $C_{\tilde{x}_{q(m_q-1)},\ldots,\tilde{x}_{q0}}$ and run L2U on the modified credal network.

Overall, the joint use of the binarization techniques described in this section, with the L2U algorithm represents a general procedure for efficient approximate updating in credal networks. Clearly, the lack of a theoretical quantification of the outer approximation provided by the binarization as in Theorem 8, together with the fact that the posterior probabilities computed by L2U can be lower as well as upper approximations, suggests the opportunity of a numerical investigation of the quality of the overall approximation, which is the argument of the next section.

5.1.5 Numerical Tests

We have implemented a *binarization algorithm* to binarize credal networks as in Definition 6 and run experiments for two sets of 50 random credal networks based on the topology of the *Alarm* network [BSCC89]. The binarized networks were updated by an implementation of L2U, choosing the node "VentLung", which is a binary node, as target variable, and assuming no evidences. The L2U algorithm converges after 3 iterations and the overall computational time is quick: posterior beliefs for the networks were produced in less than one second in a Pentium computer, while the exact calculations used for the comparisons, based on branch-and-bound techniques [dCC04], took a computational

time between 10 and 25 seconds for each simulation.[6] Results can be viewed in Figure 5.2.

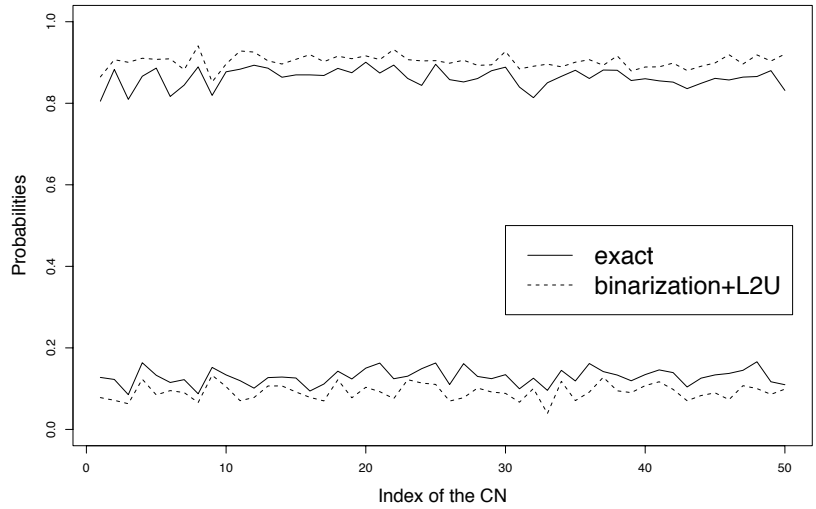
As a comment, we note a good accuracy of the approximations with a mean square error around 3% and very small deviations. Remarkably the quality of the approximation is nearly the same for both the sets of simulations. Furthermore, we observe that the posterior intervals returned by the approximate method always include the corresponding exact intervals. This seems to suggest that the approximation due to the binarization dominates that due to L2U. It should also be pointed out that the actual difference between the computational time required by the two approaches would dramatically increase for larger networks: the computational complexity of the branch-and-bound method used for exact updating is exponential in the input size, while both our binarization algorithm and L2U (assuming that it converges) take a linear time; of course both the approaches have an exponential increase with an increase in the number of categories for the variables.

5.2 Exact Binarization & GL2U

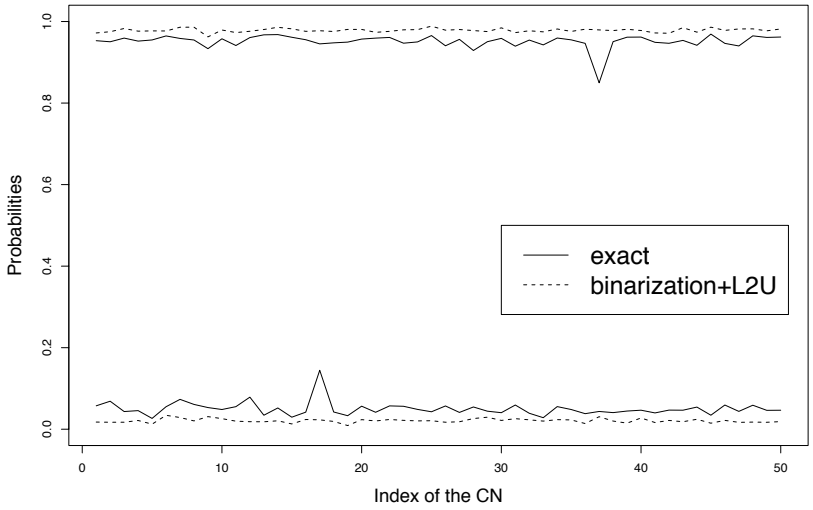
In this section we improve the "binarization + L2U" updating procedure considered in the previous section. Such improvement is based on an important representation result: that any credal network can be equivalently reformulated as one with binary variables. The corresponding transformation, which is considerably more complex than in the Bayesian case, is based on two distinct transformations: (i) first we reformulate the original credal network as a separately specified model over a wider domain by means of its decision-theoretic specification according to the procedures described in Sections 4.1 and 4.2, (ii) then, we apply the binarization described in Section 5.1.4.

We prove that the sequential application of these two transformations, originally developed for independent reasons, returns an equivalent binary representation of the original credal network (Section 5.2.1). Such equivalent binary credal network can be finally updated by L2U. Overall, that leads to a *generalized loopy 2U* (GL2U) algorithm for the updating in general credal networks, whose only source of approximation is the loopy part (Section 5.2.2). The algorithm, which takes polynomial time (Section 5.2.3), has been implemented in a software tool. Experimental evaluations in Section 5.2.4 show that its accuracy is comparable to that of state-of-the-art approximate methods for credal networks. This, together with its scalability, makes of GL2U the algorithm of choice especially for large nets.

[6] The exact inferences have been computed by Cassio Polpo de Campos.



(a) Conditional credal sets with 4 vertices



(b) Conditional credal sets with 10 vertices

Figure 5.2: A comparison between the exact results and approximations returned by the "binarization+L2U" procedure for the upper and lower values of $P(\text{VentLung} = 1)$ on two sets of 50 randomly generated credal networks based on the ALARM, with a fixed number of vertices for each conditional credal set.

5.2.1 Exact Binarization

In this section, we consider the sequential application of the transformations detailed in Section 4.1 and Section 5.1. Thus, given a credal network $\langle \mathcal{G}, \mathbf{P}(\mathbf{X}) \rangle$, we first obtain $\langle \mathcal{G}', \mathbf{P}'(\mathbf{X}') \rangle$ by a decision-theoretic specification, and hence, after the binarization, $\langle \tilde{\mathcal{G}}', \tilde{\mathbf{P}}'(\tilde{\mathbf{X}}') \rangle$. The latter credal network is called *exact binarization* of the first. Such a terminology is justified by the following result.

Theorem 9. *Consider a credal net $\langle \mathcal{G}, \mathbf{P}(\mathbf{X}) \rangle$ and its exact binarization $\langle \tilde{\mathcal{G}}', \tilde{\mathbf{P}}'(\tilde{\mathbf{X}}') \rangle$. Let $K(\mathbf{X})$ and $\tilde{K}'(\tilde{\mathbf{X}}')$ be their corresponding strong extensions. Then:*

$$K(\mathbf{X}) = \tilde{K}'(\tilde{\mathbf{X}}), \tag{5.15}$$

where $\tilde{K}'(\tilde{\mathbf{X}})$ is obtained marginalizing out of $\tilde{K}'(\tilde{\mathbf{X}}')$ the variables in $\tilde{\mathbf{X}}' \setminus \tilde{\mathbf{X}}$.

In order to prove Theorem 9, we first need the following result.

Lemma 2. *Consider a CN made of a single node X with vacuous $K(X) := K_{\Omega_X^*}(X)$, where $\Omega_X^* \subseteq \Omega_X$. Let $\tilde{K}(\tilde{X})$ denote the strong extension of its binarization (as described in Section 5.1). Then:*

$$\tilde{K}(\tilde{X}) = K(X). \tag{5.16}$$

Proof. Let $\tilde{d} := \log_2 |\Omega_X|$ and $\tilde{X} := (\tilde{X}^0, \ldots, \tilde{X}^{\tilde{d}-1})$. Consider a generic $\tilde{P}_*(\tilde{X}) \in \text{ext}[\tilde{K}(\tilde{X})]$. As described in Section 5.1, a corresponding mass function over X, say $P_*(X) := \tilde{P}_*(\tilde{X})$, can be therefore defined. The following factorization holds:

$$\tilde{P}_*(\tilde{x}) = \prod_{j=0}^{\tilde{d}-1} \tilde{P}_*(\tilde{x}^j | \tilde{x}^{j-1}, \ldots, \tilde{x}^0), \tag{5.17}$$

for each $\tilde{x} \in \Omega_{\tilde{X}}$ such that $(\tilde{x}^0, \ldots, \tilde{x}^{\tilde{d}-1}) = \tilde{x}$. For each j=0, $\ldots$, $\tilde{d} - 1$ and each possible value of their parents, the conditional mass functions $\tilde{P}_*(\tilde{X}^j | \tilde{x}^{j-1}, \ldots, \tilde{x}^0)$ are vertices of their corresponding conditional credal sets because of Proposition 1. Thus, the values of the conditional probabilities on the right-hand side of Equation (5.17) are obtained by a minimization as in Equation (5.6) (or an analogous maximization). The values to be minimized are obtained from Equation (5.1), where the conditional probabilities on the right-hand side are the vertices of $K(X)$, i.e., the $m := |\Omega_X^*|$ degenerate extreme mass functions of the vacuous credal set $K_{\Omega_X^*}(X)$. This means that there is only a non-zero term in the sum in Equation (5.1) and therefore each vertex of $K_{\Omega_X^*}$ produces a degenerate conditional mass function for the corresponding binary variable. Consequently, also the extreme values returned by Equation (5.6) will be degenerate. We can therefore conclude that, according to Equation (5.17), also $\tilde{P}_*(\tilde{X})$ and hence

$P_*(X)$ is a degenerate mass functions. Let $x_* \in \Omega_X$ be the state of X such that $P_*(X = x_*) = 1$. Considering Equation (5.17) for $\tilde{x}_* \in \Omega_{\tilde{X}}$, we conclude that all the conditional probabilities on the right-hand side are equal to one. Considering the highest order bit, according to Equation (5.1) and denoting by $P_k(X)$ a vertex of $\Omega_*(X)$, we have $\tilde{P}_*(\tilde{x}_*^{\tilde{d}-1}|\tilde{x}_*^{\tilde{d}-2},\ldots,\tilde{x}_*^0) = P_k(x_*) = 1$, that requires $x_* \in \Omega_X^*$. Thus, $P_*(X) \in \text{ext}[K(X)]$, that implies $\text{ext}[\tilde{K}(\tilde{X})] \subseteq \text{ext}[K(X)]$, and finally $\tilde{K}(\tilde{X}) \subseteq K(X)$. On the other side, as an obvious corollary of Theorem 8, $\tilde{K}(\tilde{X}) \supseteq K(X)$, and hence the thesis. □

Proof of Theorem 9. Consider a generic $\tilde{P}'_*(\tilde{\mathbf{X}}') \in \text{ext}[\tilde{K}'(\tilde{\mathbf{X}}')]$. The following factorization holds:

$$\tilde{P}'_*(\tilde{\mathbf{x}}') = \prod_{i=1}^{2n}\prod_{j=0}^{\tilde{d}_i-1} \tilde{P}'_*(\tilde{x}_i^j|\tilde{\pi}_i^j) = \prod_{i=1}^{2n} \tilde{P}'_*(\tilde{x}_i^0,\ldots,\tilde{x}_i^{\tilde{d}_i-1}|\tilde{\pi}'_i), \tag{5.18}$$

for each $\tilde{\mathbf{x}}' \in \Omega_{\tilde{\mathbf{X}}'}$, where the values of the other variables are those consistent with $\tilde{\mathbf{x}}$, and the last equality is obtained through chain rule. Equation (5.18) implicitly defines $P'_*(X_i|\pi'_i) := \tilde{P}'_*(\tilde{X}_i^0,\ldots,\tilde{X}_i^{\tilde{d}_i-1}|\tilde{\pi}'_i)$. According to the discussion in Section (4.1), for each $i = 1,\ldots,n$ and $\pi_i \in \Omega_{\Pi_i}$, $K'(X_i|\pi'_i)$ is a credal set made of a single point. Thus, as an obvious corollary of Theorem 7, we have that $P'_*(X_i|\pi'_i) \in \text{ext}[K'(X_i|\pi'_i)]$, being in fact the only element of this credal set. Similarly, for each $i = 1,\ldots,n$, the conditional credal set $K'(X_{i+n}|\pi'_{i+n})$ is vacuous. Thus, regarding this conditional credal set as a CN made of a single node, we can invoke Lemma 1 and obtain from $\tilde{P}'_*(\tilde{X}_{i+n}|\tilde{\pi}'_{i+n}) \in \text{ext}[\tilde{K}'(\tilde{X}_{i+n}|\tilde{\pi}'_{i+n})]$ that $P'_*(X_{i+n}|\pi'_{i+n}) \in \text{ext}[K'(X_{i+n}|\pi'_{i+n})]$. Overall, we have proved that $P'_*(\mathbf{X}')$ is a combination of local vertices of the conditional credal sets of $\langle \mathcal{G}', \mathbf{P}'(\mathbf{X}')\rangle$. Thus, $P'_*(\mathbf{X}') \in \text{ext}[K'(\mathbf{X}')]$, from which $\text{ext}[\tilde{K}'(\tilde{\mathbf{X}}')] \subseteq \text{ext}[K'(\mathbf{X}')]$, and finally $\tilde{K}'(\tilde{\mathbf{X}}') \subseteq K'(\mathbf{X}')$. On the other side, according to Lemma 1, $\tilde{K}'(\tilde{\mathbf{X}}') \supseteq K'(\mathbf{X}')$. Thus, $\tilde{K}'(\tilde{\mathbf{X}}') = K'(\mathbf{X}')$. Marginalizing on both the sides we get $\tilde{K}'(\tilde{\mathbf{X}}) = K'(\mathbf{X})$. Finally, Theorem 4 states that $K(\mathbf{X}) = K'(\mathbf{X})$, from which the thesis. □

According to Equation (5.15), we can regard $\langle \tilde{\mathcal{G}}', \tilde{\mathbf{P}}'(\tilde{\mathbf{X}}')\rangle$ as an equivalent binary representation of $\langle \mathcal{G}, \mathbf{P}(\mathbf{X})\rangle$. A similar equivalence has been already obtained for Bayesian networks in Section 5.1.3. Let us stress that the generalization to credal networks presented here is a substantial advancement: while an equivalent binary representation of a single joint mass function (corresponding to a Bayesian network) simply consists into an appropriate relabeling of the joint states, for credal networks and hence for credal sets the exact binarization must exactly reproduce the geometrical structure of the original convex sets.[7] It is therefore noteworthy that a procedure developed independently as

[7] For this reason a straight binarization, as described in Section 5.1, does not produce an equivalent model, unless a decision-theoretic specification is done first.

the *decision-theoretic specification* was identified as the key to produce an equivalent representation.

It should be also pointed out that, even we have focused on the case of credal networks with separately specified credal sets, Theorem 9 holds also for so-called *non-separately specified* credal networks, for which a decision theoretic specification can be provided as well. For the same reason, the algorithm presented in the following section can be equivalently applied to any, separately or non-separately specified, credal network and is therefore very general.

5.2.2 GL2U

Theorem 9 can be regarded as a result with self-sufficient relevance for representation issues. Moreover, it is a basis for the solution of general inference problems, as stated by this straightforward corollary.

Corollary 4. *Any inference problem on a credal network can be equivalently computed in its exact binarization.*

According to Corollary 4, we can therefore consider a so-called *generalized* L2U algorithm (GL2U), where given an updating problem on a credal network, we solve by L2U the corresponding updating problem on the exact binarization of the original credal network. The overall procedure is still approximate, but differently from the procedure without decision-theoretic specification considered in the previous section, the only source of approximation is the loopy component.

5.2.3 Complexity Issues

Consider the original credal net before any transformation. Let:

$$\bar{\omega} := \max_{i=1}^{n} |\Omega_{X_i}|, \tag{5.19}$$

$$\bar{\pi} := \max_{i=1}^{n} |\Pi_i|, \tag{5.20}$$

$$\bar{k} := \max_{i=1}^{n} \Big| \bigcup_{\pi_i \in \Omega_{\Pi_i}} \text{ext}[K(X_i|\pi_i)] \Big|. \tag{5.21}$$

These are, respectively, the worst-case number of states, parents, and vertices, over the variables in the net. We assume that there is a variable in the net, say X_j, which attains all the three worst cases. This implies that in the exact binarization, the variable $\tilde{X}_{j+n}^{\tilde{d}_{j+n}-1}$ maximizes the number of incoming arcs; call such a maximum $\bar{a}$. Each of the parents of X_j is transformed in a cluster of

$\lceil \log_2 \bar{\omega} \rceil$ binary nodes, which contribute then with $\bar{\pi} \lceil \log_2 \bar{\omega} \rceil$ incoming arcs. The node $\tilde{X}_{j+n}$ in the decision-theoretic specification is instead transformed into a cluster of $\lceil \log_2 \bar{k} \rceil$ binary nodes, which contribute with additional $\lceil \log_2 \bar{k} \rceil - 1$ arcs (the last one does not contribute as it is just the node on which we focus, i.e., $\tilde{X}_{j+n}^{\tilde{d}_{j+n}-1}$).

Overall, we obtain that $\bar{a} = \bar{\pi} \lceil \log_2 \bar{\omega} \rceil + \lceil \log_2 \bar{k} \rceil - 1$. By applying 2U, the worst-case complexity local to node $\tilde{X}_{j+n}^{\tilde{d}_{j+n}-1}$ is then $O(2^{2\bar{a}}) = O((\bar{\bar{\omega}}^{\bar{\pi}} \bar{\bar{k}})^2)$ (see [FZ98, Section 5], where we use the extra upper bar to denote the smallest power of 2 that is larger than or equal to the considered number. This is the worst-case complexity local to a node. Globally, an iteration of GL2U is linear in the size (i.e., the longest path) of the net, which can be regarded as a linear function of the size of the original network.

5.2.4 Numerical Tests

In order to test the performance of GL2U, we have chosen two well-known nets: *Alarm* [BSCC89] and *Insurance* [BKRK97], as well as some random generated nets. We work with random polytrees with 50 nodes (Polyt-50), and random multiply connected nets with 10 and 25 nodes (Multi-10 and Multi-25, respectively). For the Alarm and the Insurance nets, we use the original graph (37 and 27 nodes, respectively) and the original number of possible states for each variable. Ten nets are generated with random parameterizations and two vertices in each local credal set. The same is repeated with four instead of two vertices. For the random polytree nets, we generate random graphs with 50 nodes and at most 4 categories in each variable. Ten nets with two vertices and ten nets with four vertices by local credal set are created. With random multiply connected nets, we work with 10 and 25 nodes, and 4 and 8 categories by variable. Again, ten nets are used in each configuration.[8]

In each net, we run marginal inferences for each one of its variables using GL2U, the "rough" binarization without decision-theoretic specification as proposed in the previous section (BIN), the state-of-the-art approximate local search method described in [dRCdC03] (LS) limited to 20 iterations in order to have running times similar to those of GL2U, and the exact method presented in [dCC07]. Table 5.1 shows the mean square error of LS, GL2U and BIN methods when compared to the exact solution. We point out that we have always observed convergence of GL2U (and BIN).

We verify that GL2U always display convergenge after a small number of

[8] The simulations presented in this section has been done in cooperation with Cassio Polpo de Campos.

		LS	GL2U	BIN
Multi-10	4 / 2	0.0189	**0.0140**	0.0181
Multi-10	8 / 2	0.0195	**0.0107**	0.0338
Multi-10	4 / 4	**0.0120**	0.0175	0.0308
Multi-10	4 / 8	**0.0027**	0.0125	0.0222
Multi-10	8 / 4	0.0234	**0.0189**	0.0693
Multi-25	4 / 2	0.0231	**0.0160**	0.0184
Multi-25	4 / 4	0.0248	**0.0204**	0.0303
Polyt-50	4 / 2	**0.0112**	0.0193	0.0289
Polyt-50	4 / 4	**0.0145**	0.0221	0.0392
Insurance	5 / 2	**0.0055**	0.0117	0.0175
Insurance	5 / 4	**0.0113**	0.0132	0.0193
Alarm	4 / 2	0.0290	**0.0190**	0.0302
Alarm	4 / 4	0.0331	**0.0239**	0.0423

Table 5.1: Average mean square error of LS, GL2U and BIN methods on several nets. The second column reports the maximum number of states and the maximum number of vertices for each conditional credal set. For each row, the smallest error is boldfaced.

iterations and improves, often substantially, the approximation accuracy when compared to the straight binarization BIN; moreover, it has accuracy similar to LS. Moreover, the running time and the amount of allocated memory for LS rapidly increases with the size of the net, which makes unfeasible a solution for large nets, which can be instead quickly updated by GL2U (see Figure 5.3).

5.3 Summary and Outlooks

In this chapter we have proposed an efficient, accurate, and scalable algorithm for approximate updating on credal nets. This task is achieved augmenting the credal net by a number of nodes enumerating the extreme points of the conditional credal sets and then transforming the credal net in a corresponding credal net over binary variables, and updating such binary credal net by the loopy version of 2U. Remarkably, the procedure can be applied to any credal net, without restrictions related to the net topology or to the number of possible states of the variables, and the only approximation is due to the loopy propagation.

Empirical analysis show that the algorithm can be regarded as a state-of-the-art procedure for approximate inference in credal nets both in terms of accuracy and scalability. The algorithm is also purely distributed and allows for simultaneous updating of all the variables in the net: these characteristics are usually not

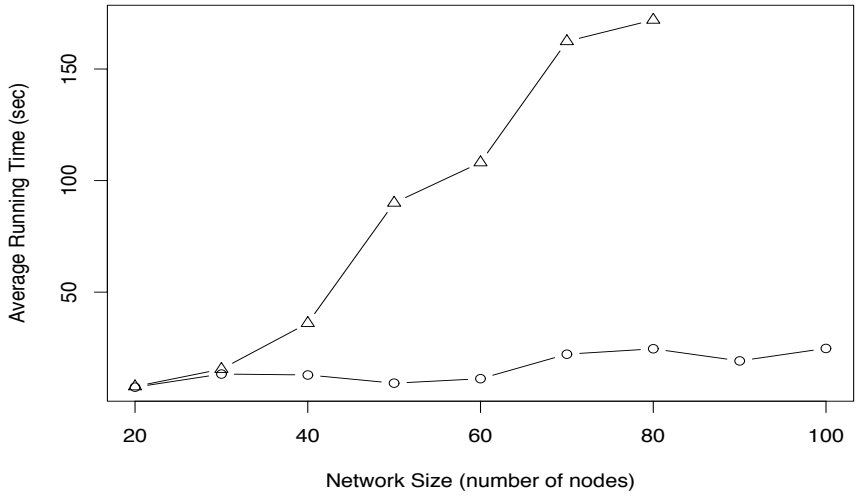


Figure 5.3: Average running time versus net size for LS (triangles) and GL2U (circles). LS cannot solve CNs with more than 80 nodes for memory constraints.

shared by optimization-based algorithms for CNs, and appear especially suited for complex applications (e.g., sensor networks). Moreover, the computational complexity GL2U makes it possible to solve (very) large networks, which can either not be updated by existing algorithms or not as accurately as GL2U.

As a future work, we intend to perform other numerical studies of the performance of the algorithm, also with extensive comparisons with other updating algorithms for credal networks. We also intend to specialize L2U to the updating of binary credal networks obtained through the exact binarization of a generic credal networks. The particular features of these specific credal nets, allow for an improvement of the performances of L2U and hence of the overall computational time.

Chapter 6

Fast Algorithms for Robust Classification with Bayesian Nets

In this chapter, we focus on a well-known classification task with Bayesian networks: predicting the state of a target variable given an incomplete observation of the other variables in the network, i.e., an observation of a subset of all the possible variables. To provide conclusions robust to near-ignorance about the process that prevents some of the variables from being observed, we adopt *conservative updating*, which is just a special case of CIR, corresponding to a situation where all the missing observations are missing in a not-MAR way. We address the problem to efficiently compute the conservative updating rule for robust classification with Bayesian networks. We show first that the general problem is *NP-hard*, thus establishing a fundamental limit to the possibility to do robust classification efficiently. Then we define a wide subclass of Bayesian networks that does admit efficient computation. We show this by developing a new classification algorithm for such a class, which extends substantially the limits of efficient computation with respect to the previously existing algorithm. The algorithm is formulated as a *variable elimination* procedure, whose computation time is linear in the input size.

6.1 Preliminaries

Probabilistic expert systems yield conclusions on the basis of *evidence* about a domain. For example, we have seen how Bayesian networks are queried for updating the confidence on a target variable given an evidence, i.e., after observing the value of other variables in the network model. Very often, at the time of a query, only a subset of all the variables is in a known state, as there is a so-called *missingness process* that prevents some variables from being observed. This is

a crucial point. The traditional way to update beliefs in probabilistic expert systems relies on Kolmogorov's conditioning rule. In order to yield correct conclusions, such a rule needs that the missingness process is explicitly modeled, or at least that it does not act in a selective way (i.e., that it is not malicious in producing the missingness). Unfortunately, the missingness process may be difficult to model, and assuming that it is unselective is equivalent to assuming the well-known *missing at random* (MAR) condition [LR87], which is often unrealistic [GH03].

To address such a fundamental issue, De Cooman and Zaffalon [dCZ04] have recently derived a new rule to update probabilities with expert systems in the case of near-ignorance about the missingness process. As a more realistic model of this condition of partial information, the new, so-called, *conservative updating rule* (or CUR), yields lower and upper probabilities in general, as well as partially determined decisions. With classification problems, for instance, where the goal is to predict the state of the target variable (also called *class variable*) given an evidence, CUR leads to set-based classifications, or, in other words, to *credal classifiers* [Zaf02] (see Section 6.2.2). De Cooman and Zaffalon have indeed specialized CUR to solve classification problems with Bayesian networks. Yet, their algorithm is efficient only on a relatively limited class of Bayesian networks: those in which the *Markov blanket*[1] of the class variable together with the variable itself forms a polytree, that is, a graph that becomes a tree after dropping the orientation of the arcs. Two natural questions arise in relationship with the above algorithm: is it possible to provide an algorithm for CUR-based classification that is similarly efficient on more general network structures? And, at a more fundamental level, what are the limits of efficient computation posed by the nature of the problem?

In this chapter we address both questions. Initially, we prove the hardness of the problem, thus solving the second question: doing classification with CUR on Bayesian nets is shown to be *NP-hard* in Section 6.3. This parallels analogous results obtained for Bayesian nets that implement the traditional updating [Coo90]; in those cases, the algorithms are efficient when the entire graph is a polytree, and are exponential with more general, so called, *multiply connected graphs*.

Then we address the first question by developing a new algorithm that substantially extends the limits of efficient computation with respect to De Cooman and Zaffalon's original algorithm. We achieve this goal, which is relatively involved from the technical point of view, in different steps. We first introduce in Section 6.4.1 a new kind of network model, called *s-network*, that abstracts

[1]The set of nodes made by the parents, the children, and the parents of the children of a given variable.

the main features of a CUR-based classification on a Bayesian net. Secondly, in Section 6.4.2, we show that our classification problem can be solved through suitable calculations on a corresponding s-network; an algorithm that implements this mapping is also provided. In Section 6.5.1 this particular calculation over s-networks is proved to be equivalent to a *variable elimination* procedure in the abstract framework of a *valuation algebra* [Koh03] (see Section 6.2.3). In Section 6.5.2, a strategy that defines a particular order in which the variables should be eliminated is provided for the special case of classification problems such that the corresponding s-network is a polytree (or a collection of them, i.e., a *polyforest*). In this way it is possible to provide a linear time algorithm performing these calculations (Section 6.5.3). That concerns also many cases when the class variable with its Markov blanket forms a multiply connected graph in the original Bayesian net. This, together with the fact that the complexity of CUR-based classification depends on the structure of the Markov blanket rather than that of the entire net, makes the new algorithm efficient on a truly large subset of Bayesian networks.

Overall, we develop a computational basis to do classification in expert systems when there is little knowledge about the process producing the missingness. This enables efficient computation to take place on a large subset of Bayesian networks, which is of course important for applications. General remarks about CUR-based classification are in Section 6.6.

6.2 Setup

6.2.1 Classification by Bayesian Networks

In this chapter we adopt a slightly different formalism for Bayesian networks, which is more oriented to classification problems. Thus, we consider the random variables $A_0, \ldots, A_n$, where variable A_k $(k = 0, \ldots, n)$ takes generic value a_k from the finite set $\mathscr{A}_k$. The available information about the relationship between the random variables is specified by a (prior) mass function $P(A_0, \ldots, A_n)$, which we assume to be positive in the following.

The mass function $P(A_0, \ldots, A_n)$ can be conveniently provided by a domain expert using a Bayesian network. Accordingly, each node A_k holds a conditional mass function $P(A_k|\pi_{A_k})$ for each joint state π_{A_k} of its direct predecessor nodes (or *parents*) Π_{A_k}. The joint probability $P(a_0, \ldots, a_n)$ is given by $P(a_0, \ldots, a_n) = \prod_{k=0}^{n} P(a_k|\pi_{A_k})$ for all the $(n+1)$-tuples $(a_0, \ldots, a_n) \in \bigtimes_{k=0}^{n} \mathscr{A}_k$, where π_{A_k} is the assignment to the parents of A_k consistent with $(a_0, \ldots, a_n)$.

For our purposes, we arbitrary choose A_0 as target node, aiming at predicting its state given values of some other nodes. In the following A_0 is called *class*

variable and is also be denoted by C, with generic value c from the set of classes $\mathscr{C} := \mathscr{A}_0$. The remaining variables are called *attribute variables*, and their values *attributes*. We refer to this predictive problem as *classification*.

6.2.2 Robust Classification

In classification problems, we typically observe (or measure) only a subset of the attribute variables at the time of a query. In order to update probabilities about the class variable given the observations, there is a frequent habit to neglect the missing attribute variables after the conditioning bar. However, this method is justified only when the process responsible for the missingness is unselective, that is, when it creates the missingness without any specific purpose. More technically, this happens when the probability that a measurement is missing is the same irrespectively of the specific measurement. In this case we say that the process is MAR [LR87]. Unfortunately, MAR is quite a strong assumption [GH03] and for this reason MAR-based approaches are somewhat criticized (see also [Man03]).

Following a deliberately conservative approach, De Cooman and Zaffalon [dCZ04] have instead used *coherent lower previsions* [Wal91], which are equivalent to credal sets, to model the case of near-ignorance about the missingness process. This has led to a new rule to update beliefs in expert systems that is called conservative updating rule. In order to denote incomplete observations of the attribute variables (the class variable is clearly unobserved, as it is the variable to predict), let us use E for the subset of the attribute variables that are observed and e for their joint value. Let us denote by R the remaining attribute variables, whose values are missing. We also denote the set of their possible joint values by $\mathscr{R}$, and a generic element of that set by r. Observe that for every $r \in \mathscr{R}$, the attributes vector (e,r) is a possible *completion* of the incomplete observation $(E,R) = (e,*)$, where the symbol $*$ denotes missing values. The updated probability of the class variable given $(e,*)$ is an interval, according to the conservative updating rule, whose extremes are the following:

$$\underline{P}(c|e,*) := \min_{r\in\mathscr{R}} P(c|e,r) \tag{6.1}$$

$$\overline{P}(c|e,*) := \max_{r\in\mathscr{R}} P(c|e,r). \tag{6.2}$$

In this chapter we are concerned with predicting the value of the class variable given $(e,*)$. This is equivalent to producing the set of the *undominated* classes according to the conservative updating rule. Say that class c' *credal-dominates*, or simply *dominates*, class c'', if $P(c'|e,r) > P(c''|e,r)$ for all $r \in \mathscr{R}$. The notation $c' \succ c''$ is adopted to formalize this kind of dominance. A class

is called *undominated* if there is no class that dominates it. This dominance criterion is a special case of *strict preference* proposed by Walley [Wal91, Section 3.7.7]. In other words, the conservative updating rule generally produces set-based classifications, where each class in the output set should be regarded as a candidate *optimal* class. Classifiers that produce set-based classifications are also called *credal classifiers* by Zaffalon [Zaf02].

It is easy to show that testing whether $c' \succ c''$ can be carried out in the following equivalent way:

$$\min_{r \in \mathscr{R}} \frac{P(c', e, r)}{P(c'', e, r)} > 1. \tag{6.3}$$

Let us use π' and π'' to denote values of parent variables consistent with the completions (c', e, r) and (c'', e, r), respectively. Regarding C, let π denote the value of its parents consistent with (e, r). Furthermore, without loss of generality, let $A_1, \ldots, A_m$, $m \leq n$, be the *children* (i.e., the direct successor nodes) of C. Denote by B^+ the union of C with its Markov blanket. De Cooman and Zaffalon [dCZ04] show that the minimum in (6.3) can be computed by restricting the attention to B^+, in the following way:

$$\min_{\substack{a_j \in \mathscr{A}_j, \\ A_j \in B^+ \cap R}} \left[\frac{P(c'|\pi_C)}{P(c''|\pi_C)} \prod_{i=1}^{m} \frac{P(a_i|\pi'_{A_i})}{P(a_i|\pi''_{A_i})} \right]. \tag{6.4}$$

Note that Expression (6.4) does not change by removing the arcs such that their second endpoint[2] is neither C nor one of its children. In the following, we refer to B^+ just as the subgraph deprived of those negligible arcs.

6.2.3 Local Computations on Valuation Algebras

Many different formalisms for managing uncertainty in expert systems share a common algebraic structure based on elementary operations such as *aggregation* of knowledge and *focus* on part of the overall information. In this section we present a formal definition of this structure together with an algorithm for solving many computational tasks on this framework.

Let $\mathscr{V}$ be a finite collection of random variables over finite domains and Φ a set of abstract objects called *valuations*. Three operations are assumed to be defined over Φ and $\mathscr{V}$, namely a *labeling* $d : \Phi \to 2^{\mathscr{V}}$, a *combination* $\otimes : \Phi \times \Phi \to \Phi$ and a *variable elimination* $\mathrm{El} : \Phi \times \mathscr{V} \to \Phi$.

Every valuation $\phi \in \Phi$ is interpreted as a piece of knowledge about the possible values of the variables in $d(\phi) \subseteq \mathscr{V}$ and $d(\phi)$ is called *domain* of ϕ.

[2] Two nodes connected by an arc are called its *endpoints*. The first endpoint is the node from which the arc departs, while the second is the remaining node.

Given two valuations $\phi, \psi \in \Phi$, the *combined valuation* $\phi \otimes \psi$ represents the aggregate knowledge coming from both ϕ and ψ. Finally, given a valuation $\phi \in \Phi$ and a variable $A \in \mathcal{V}$, $\mathrm{El}(\phi, A)$ represents a valuation that focuses on the knowledge associated to ϕ with no attention to what is related to A. We use also the notation ϕ^{-A} to denote $\mathrm{El}(\phi, A)$.

The 5-tuple $(\Phi, \mathcal{V}, d, \otimes, \mathrm{El})$ is called *valuation algebra* (VA) [Koh03] if the operations of labelling d, combination $\otimes$, and variable elimination El over the set of valuations Φ and the set of random variables $\mathcal{V}$, satisfy the following system of axioms:

(A1) Φ is commutative and associative under $\otimes$

(A2) If $\phi, \psi \in \Phi$, then $d(\phi \otimes \psi) = d(\phi) \cup d(\psi)$

(A3) If $\phi \in \Phi$ and $V \in \mathcal{V}$ is such that $V \in d(\phi)$, then $d(\phi^{-V}) = d(\phi) \setminus \{V\}$

(A4) If $\phi \in \Phi$ and $V, W \in \mathcal{V}$ then $(\phi^{-V})^{-W} = (\phi^{-W})^{-V}$

(A5) If $\phi, \psi \in \Phi$ and $V \in \mathcal{V}$ is such that $V \notin d(\phi)$, then $(\phi \otimes \psi)^{-V} = \phi \otimes \psi^{-V}$.

Let $(\Phi, \mathcal{V}, d, \otimes, \mathrm{El})$ be a valuation algebra and $\{\phi_i\}_{i=0}^m$ a set of valuations in Φ such that $\bigcup_{i=0}^m d(\phi_i) = \mathcal{V}$. According to (A1), a *joint valuation* $\phi := \otimes_{i=0}^m \phi_i$ of this set can be defined with no ambiguities. According to (A2), $d(\phi) = \mathcal{V}$. According to (A4), the valuation obtained eliminating from ϕ all the variables of its domain is independent from the elimination sequence and can therefore be unequivocally denoted as $\phi^{-\mathcal{V}}$. According to (A3), $\phi^{-\mathcal{V}}$ is a valuation with empty domain and is called the *full marginal* of the joint valuation ϕ.

The complexity of the operation of variable elimination typically increases exponentially with the domain of the valuation considered. That often makes the computation of $\phi^{-\mathcal{V}}$ intractable, even if all the given valuations are defined on small domains.

However, (A5) suggests the possibility of eliminating some variable on a local domain, that is, without explicitly computing the joint valuation. This approach, called *fusion algorithm* [Koh03], consists in the elimination of a variable only from the combination of the valuations such that the variable to eliminate is in their domain, i.e.:

$$\phi^{-V} = \left(\otimes_{i=0,\dots,m/V \notin d(\phi_i)} \phi_i \right) \otimes \left(\otimes_{j=0,\dots,m/V \in d(\phi_j)} \phi_j \right)^{-V}. \tag{6.5}$$

The elementary procedure portrayed in (6.5) can be iterated over all the elements of $\mathcal{V}$, leading to $\phi^{-\mathcal{V}}$. According to (A4), any elimination sequence can be employed. Nevertheless, it should be pointed out how different sequences require in general different computational times.

6.3 Hardness of CUR-Based Classification

Call CCUR the problem to compute the undominated classes in a CUR-based classification problem with Bayesian nets. Let us initially focus on the binary version of the CCUR problem, that is, on a classification problem with only two classes, say c' and c''. We denote by CCURD the corresponding decision problem that involves deciding whether or not c' dominates c''. CCURD is clearly equivalent to (6.3), being 'true' (T) if (6.4) is greater than one and 'false' (F) otherwise. As a preliminary result, we prove that CCURD is *coNP-complete*, i.e., the complement of an *NP-complete* problem [Pap94]. In our proof, we take inspiration from the well-known result of Cooper [Coo90], concerning probabilistic inference with Bayesian nets.

Recall that a decision problem $\mathscr{Q}$ is NP-complete if $\mathscr{Q}$ lies in the class NP and some known NP-complete problem $\mathscr{Q}'$ polynomially transforms to $\mathscr{Q}$ [GJ79, p. 38]. In our case, we transform a well known NP-complete problem, called 3-satisfiability (3SAT) [GJ79], to the complement of CCURD. Let us recall the definition of 3SAT.

Let $\mathscr{U}$ be a collection of n Boolean variables. If U is a variable in $\mathscr{U}$ then u and $\neg u$ are called *literals* over $\mathscr{U}$. The literal u is true if and only if the variable U is true, while $\neg u$ is true if and only if the variable U is false. Let $\mathscr{K} = \{K_1, \ldots, K_m\}$ be a non-empty collection of *clauses*, which are disjunctions of triples of literals, corresponding to different[3] variables of $\mathscr{U}$. The collection of clauses $\mathscr{K}$ over $\mathscr{U}$ is called *satisfiable* if and only if there exists a *truth assignment* for $\mathscr{U}$, that is, an assignment of Boolean values to the variables in $\mathscr{U}$, such that all the clauses in $\mathscr{K}$ are simultaneously true. The 3SAT decision problem involves determining whether or not there is a truth assignment for $\mathscr{U}$ such that $\mathscr{K}$ is satisfiable.

The NP-completeness of 3SAT can be used to prove the following:

Theorem 10. *CCURD is coNP-complete.*

Proof. Given a generic 3SAT instance, $\mathscr{U} = \{U_1, \ldots, U_n\}$ and $\mathscr{K} = \{K_1, \ldots, K_m\}$, we construct a Bayesian network such that $c' > c''$ if and only if $\mathscr{K}$ is not satisfiable. The nodes of the network correspond to the variables in $\mathscr{U}$, the clauses in $\mathscr{K}$ and the class C. The nodes corresponding to the clauses have four incoming arcs, three from the variables associated to the literals present in the definition of the clause and the fourth from the class node. The directed acyclic graph underlying the Bayesian network is therefore $\mathscr{G}(\mathscr{V}, \mathscr{E})$, with

[3] This assumption is not included in the original transformation of the prototypical NP-complete problem SAT to 3SAT. Nevertheless, the transformation (see for example [GJ79, p. 48]) does not require any clause to include literals corresponding to the same variable. Thus, also this version of 3SAT is NP-complete.

$\mathcal{V} = \{C, U_1, \ldots, U_n, K_1, \ldots, K_m\}$ and

$$\mathcal{E} = \{(U_{\alpha_{ij}}, K_j) \mid \begin{matrix} i = 1,2,3, \\ j = 1,\ldots,m \end{matrix} \} \cup \{(C, K_j) \mid j = 1, \ldots, m\}, \tag{6.6}$$

where α_{ij} indexes the element of $\mathcal{U}$ corresponding to the i-th literal of the clause K_j. As an example, Figure 6.1 reports the graph corresponding to a 3SAT instance with three clauses and four variables in $\mathcal{U}$.

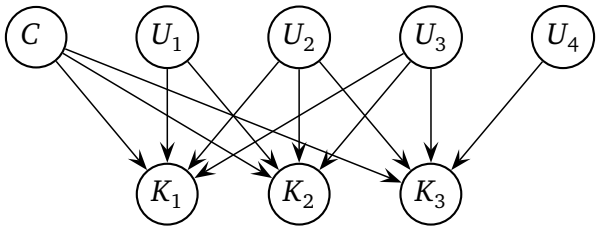


Figure 6.1: A Bayesian network corresponding to an instance of the 3SAT problem with $\mathcal{U} = \{U_1, U_2, U_3, U_4\}$ and $\mathcal{K} = \{(u_1 \vee u_2 \vee u_3), (\neg u_1 \vee \neg u_2 \vee u_3), (u_2 \vee \neg u_3 \vee u_4)\}$.

Each node of $\mathcal{G}$ is assumed to represent a Boolean variable. The unconditional mass functions for the root nodes (i.e., the nodes without incoming arcs) are assumed to be uniform. Regarding the conditional mass functions we define them as in Table 6.1. Those values define a unique positive mass function for each clause and for every possible value of the parents of the clause.

c	$u_{\alpha_{1j}} \vee u_{\alpha_{2j}} \vee u_{\alpha_{3j}}$	$P(K_j = \mathrm{T} \mid c, u_{\alpha_{1j}}, u_{\alpha_{2j}}, u_{\alpha_{3j}})$
c'	T	2^{-2}
c''	T	2^{-1}
c'	F	2^{-1}
c''	F	$2^{-(m+1)}$

Table 6.1: Implicit definition of the conditional mass functions for the clause K_j, for each $j = 0, \ldots, m$. With an abuse of notation, $u_{\alpha_{ij}}$ denotes the i-th literal of K_j.

The directed acyclic graph $\mathcal{G}$, together with the specified mass functions, defines a Bayesian network. This is equivalent to a joint mass function, which assigns positive probability to every event. With respect to the evidence $E = e$ in the network, we suppose all the clauses in $\mathcal{K}$ are instantiated to the state 'true'.

The remaining attribute variables, which are the variables in $\mathscr{U}$, are assumed to be missing. Expression (6.4) becomes:

$$\min_{\substack{u_j \in \{F,T\}, \\ U_j \in \mathscr{U}}} \prod_{i=1}^{m} \phi_i(u_{\alpha_{1i}}, u_{\alpha_{2i}}, u_{\alpha_{3i}}), \tag{6.7}$$

where, for each $i = 1, \ldots, m$,

$$\phi_i(u_{\alpha_{1i}}, u_{\alpha_{2i}}, u_{\alpha_{3i}}) := \frac{P(K_i = \mathrm{T} | c', u_{\alpha_{1i}}, u_{\alpha_{2i}}, u_{\alpha_{3i}})}{P(K_i = \mathrm{T} | c'', u_{\alpha_{1i}}, u_{\alpha_{2i}}, u_{\alpha_{3i}})}. \tag{6.8}$$

Using the values of Table 6.1, the functions in (6.8) take the form:

$$\phi_i(u_{\alpha_{1i}}, u_{\alpha_{2i}}, u_{\alpha_{3i}}) = \begin{cases} 2^{-1} & \text{if } u_{\alpha_{1i}} \vee u_{\alpha_{2i}} \vee u_{\alpha_{3i}} = \mathrm{T} \\ 2^{m} & \text{otherwise.} \end{cases} \tag{6.9}$$

According to (6.9), if a clause is satisfied, the corresponding function attains its minimum value. Thus, if 3SAT is true, there exists a truth assignment over $\mathscr{U}$ satisfying all the clauses in $\mathscr{K}$, and all the functions (6.8) in (6.7) are simultaneously minimized. The minimum (6.7) is therefore 2^{-m} and the corresponding CCURD instance is false. If 3SAT is false, for all truth assignments at least one clause is violated and the corresponding function takes the value 2^m. That makes (6.7) always greater than one, because all the remaining $m-1$ functions cannot be less than 2^{-1}. Thus, CCURD is true.

This shows that each 3SAT instance is equivalent to an instance of the complement of CCURD; and we have achieved this by a transformation that is polynomial in the size of the 3SAT instance. Note, in addition, that the complement of CCURD is also in the class NP. A nondeterministic algorithm to solve the complement of CCURD has only to return a truth assignment for $\mathscr{U}$, provided that the corresponding value of the functions in (6.9) can be evaluated efficiently. It follows that the complement of CCURD is NP-complete and hence the thesis. □

As a direct consequence of Theorem 10, we can prove the following:

Corollary 5. *CCUR is NP-hard.*

Proof. Let CCURD′ be the complement of CCURD. In order to prove the hardness of CCUR we consider a polynomial-time Turing reduction [GJ79, p. 111] from CCURD′ to the binary version of CCUR. Suppose a hypothetical algorithm that solves instances of the binary CCUR problem is available. Let I be a CCURD′ instance that is true if c' does not dominate c'' and false otherwise. In order to solve such an instance we use the above algorithm for CCUR problems in the

following way. If the algorithm yields c', then necessarily $c'>c''$, and I is false. If it yields both c' and c'', c' cannot dominate c'' and I is true. Analogously, if the algorithm yields only c'', I is still true. In any case, it turns out that a single call of the algorithm makes it possible to solve the CCURD′ instance I. Therefore CCURD′, which is NP-complete because of Theorem 10, is Turing reducible to the binary version of CCUR. This means that the binary version of CCUR is NP-hard, and, as a consequence, so is the general version. □

6.4 S-Networks

The hardness result of the previous section establishes a limit to the possibility to compute classifications efficiently with CUR on Bayesian nets. Yet, efficient computation is possible on special classes of Bayesian networks: in fact, De Cooman and Zaffalon [dCZ04] provide a linear time algorithm to solve CCUR problems when the subgraph B^+, defined at the end of Section 6.2.2, is singly connected. In this chapter we substantially extend such a result by providing a linear time algorithm that works in many cases also when B^+ is multiply connected.

The development of the new algorithm relies on the definition of a new kind of graphical model, called *s-network*, which allows us to abstract the main components of a CCUR problem.

6.4.1 Basic Definitions

Definition 1. *Let $\mathcal{G}$ be a directed acyclic graph in which some nodes, say $A_0,\ldots,A_m$ ($m \geq 0$), are marked as* special nodes *(or* s-nodes*) such that every arc of $\mathcal{G}$ has a special node as second endpoint. Each node of $\mathcal{G}$ is identified with a variable that takes finitely many values. Every special node A_i in $\mathcal{G}$ ($i = 0,\ldots,m$) is associated with a so-called potential $\phi_i(A_i^+)$, defined for all the values of its argument. A_i^+ is the vector variable (A_i, Π_{A_i}), with generic value a_i^+, where Π_{A_i} are the parents of A_i. The graph $\mathcal{G}$, together with the collection of potentials $\{\phi_i\}_{i=0}^m$, is called* s-network*.*

Given an s-network $\mathcal{G}$, its *minimum* is defined by

$$\min_{\substack{a_j^+ \in \mathscr{A}_j^+, \\ j\in\{0,\ldots,m\}}} \prod_{i=0}^{m} \phi_i(a_i^+). \tag{6.10}$$

Note that Definition 1 does not exclude the case of disconnected s-networks. If $\mathcal{G}_k$ is a connected component of a (disconnected) s-network $\mathcal{G}$, we can regard $\mathcal{G}_k$, together with the potentials of $\mathcal{G}$ corresponding to the s-nodes of $\mathcal{G}_k$, as an s-(sub)network. The following result holds:

Theorem 11. *Let $\mathcal{G}$ be a disconnected s-network. The minimum of $\mathcal{G}$ factorizes in the product of the minima of the s-networks corresponding to the connected components of $\mathcal{G}$ with at least one s-node.*

In order to prove Theorem 11, we first need the following result.

Lemma 3. *Let A_k and A_l be two distinct s-nodes of an s-network $\mathcal{G}$. A_k^+ and A_l^+ can share some variables if and only if A_k is a parent or a child or a sibling of A_l.*

Proof. Let S be a variable included both in $A_k^+ = (A_k, \Pi_{A_k})$ and $A_l^+ = (A_l, \Pi_{A_l})$. We distinguish the four possible cases: (i) $S = A_k = A_l$. (ii) $S = A_k$ and $S \in \Pi_{A_l}$ (iii) $S = A_l$ and $S \in \Pi_{A_k}$ (iv) $S \in \Pi_{A_k}$ and $S \in \Pi_{A_l}$.

The first case cannot take place because A_k and A_l are assumed to be distinct nodes. In the second case A_k is clearly a parent of A_l, while, *vice versa*, A_l is parent of A_k in the third case. Finally, A_k and A_l are sibling through their common parent S in the fourth case.

On the other hand, if A_k is a parent (child) of A_l, clearly A_k^+ shares the variable A_k (A_l) with A_l^+. Finally, if A_k and A_l are siblings, their common parents appear both in A_k^+ and A_l^+. □

Proof of Theorem 11. Let $(\mathcal{G}_1, \ldots, \mathcal{G}_s)$ be the connected components of $\mathcal{G}$ with at least one s-node. We denote as M_i the vector of the indices of the s-nodes that are in $\mathcal{G}_i$ ($i = 1, \ldots, s$). Clearly, $(M_1, \ldots, M_s)$ represents a partition of $M := \{0, \ldots, m\}$.

For each $k \in M_i$ and $l \in M_j$ ($i, j = 1, \ldots, s$ and $i \neq j$), A_k^+ and A_l^+ cannot share any variable because of Lemma 3. The minimum of $\mathcal{G}$ can therefore be expressed as a product of local minima $\prod_{k=1}^{s} \mu_k$, where, for each $k = 1, \ldots, s$:

$$\mu_k := \min_{\substack{a_j^+ \in \mathcal{A}_j^+, \\ j \in M_k}} \prod_{i \in M_k} \phi_i(a_i^+). \tag{6.11}$$

But (6.11) is the minimum of the s-(sub)network $\mathcal{G}_k$, that proves the thesis. □

In the next section, we show that by calculating the minima of s-networks we can solve CCUR instances.

6.4.2 Minima of S-Networks Solve CCURD Problems

Let I be a CCURD instance that involves deciding whether or not $c' > c''$. We denote by $\mathcal{G}_I$ the directed graph obtained from B^+ marking as special $C = A_0$ together with its children, removing the arcs that leave C and the observed nodes, and removing the observed nodes that are not special. The following algorithm is an obvious (linear time) implementation of this transformation:

Algorithm 1. *An algorithm to build up a graph $\mathscr{G}_I(\mathscr{V},\mathscr{E})$ given a CCURD (or CCUR) instance I.* $\mathrm{T}(\epsilon)$ *represents the first endpoint of the arc* ϵ*, while* E *is the subset of the observed attribute variables of* I*.*

```
1  G_I := B+;
2  for each V ∈ V {
3      if  V = C or C parent of V {
4          mark V as special; }}
5  for each ε ∈ E {
6      if T(ε) ∈ E or T(ε) = C {
7          remove ε; }}
8  for each V ∈ E {
9      if V not special {
10         remove V; }}
```

Each node of $\mathscr{G}_I$ is identified with a variable that takes finitely many values, as follows. The target node A_0 and the nodes of $\mathscr{G}_I$ corresponding to the observed attribute variables of I are assumed to be constants, i.e., their possibility spaces contain a single value, while the remaining nodes, which are the missing attribute variables in I, are identified with the same categorical variables of the original problem. Finally, we set:

$$\phi_0(a_0^+) := \frac{P(c'|\pi_C)}{P(c''|\pi_C)} \tag{6.12}$$

$$\phi_i(a_i^+) := \frac{P(a_i|\pi'_{A_i})}{P(a_i|\pi''_{A_i})} \qquad i = 1,\ldots,m. \tag{6.13}$$

The graph $\mathscr{G}_I$ together with the potentials as in (6.12) and (6.13) can be easily recognized to be an s-network. The computation of the minimum of this s-network solves the original CCURD instance, according to the following:

Theorem 12. *I is true if and only if the minimum of the s-network $\mathscr{G}_I$ is greater than one.*

Proof. Using (6.12) and (6.13), the minimum of $\mathscr{G}_I$ becomes:

$$\min_{\substack{a_j\in\mathscr{A}_j,\\ j=\{0,\ldots,n\}}} \left[\frac{P(c'|\pi_C)}{P(c''|\pi_C)}\cdot\prod_{i=1}^{m}\frac{P(a_i|\pi'_{A_i})}{P(a_i|\pi''_{A_i})}\right]. \tag{6.14}$$

The missing attribute variables of the CCURD instance I are exactly the non-constant variables in (6.14), while the constant variables have the same values of

the observed attribute variables in I. Finally, as observed in [dCZ04, Section 6], (6.3) is preserved by dropping the arcs leaving the nodes in the subset of the observed nodes E for each $c \in \mathscr{C}$ and $r \in \mathscr{R}$. Thus, (6.14) coincides with the expression (6.4) relative to I. That proves the thesis. □

As a numerical example, let us consider a Bayesian network over the Boolean variables $(A_0, \ldots, A_6)$ with the graphical structure displayed in Figure 6.2. Let $C := A_0$ be the class variable and c' and c'' the possible classes.

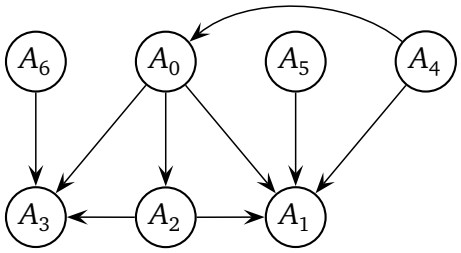


Figure 6.2: A multiply connected Bayesian network.

We assume uniform unconditional mass functions for the root nodes, while Tables 6.2, 6.3, 6.4 and 6.5 specify the conditional mass functions for the remaining nodes.

a_4	$P(C = c' \mid a_4)$
T	0.8
F	0.9

Table 6.2: Conditional mass functions for node C.

The decision whether c' dominates c'' or not, assuming all the attribute variables $(A_1, \ldots, A_6)$ to be missing, can be regarded as a CCURD instance I. First, we use Algorithm 1 to construct the graph $\mathscr{G}_I$ corresponding to the instance I. The result is the s-network displayed in Figure 6.3.

According to the procedure described in this section, each node of $\mathscr{G}_I$ is identified with the same Boolean variable of the original Bayesian network, except A_0 that is assumed to be constant. Furthermore, we can use the probability specifications in Tables 6.2, 6.3, 6.4 and 6.5 to define a potential for each special node of $\mathscr{G}_I$ as in (6.12) and (6.13). Finally, according to Theorem 12, the computation of the minimum of $\mathscr{G}_I$ solves the CCURD instance I.

c	a_2	a_4	a_5	$P(A_1 = \mathrm{T} \mid c, a_2, a_4, a_5)$
c'	T	T	T	0.4
c'	T	T	F	0.2
c'	T	F	T	0.3
c'	T	F	F	0.1
c'	F	T	T	0.7
c'	F	T	F	0.9
c'	F	F	T	0.8
c'	F	F	F	0.1
c''	T	T	T	0.2
c''	T	T	F	0.3
c''	T	F	T	0.3
c''	T	F	F	0.2
c''	F	T	T	0.4
c''	F	T	F	0.9
c''	F	F	T	0.7
c''	F	F	F	0.2

Table 6.3: Conditional mass functions for node A_1.

c	$P(A_2 = \mathrm{T} \mid c)$
c'	0.4
c''	0.7

Table 6.4: Conditional mass functions for node A_2.

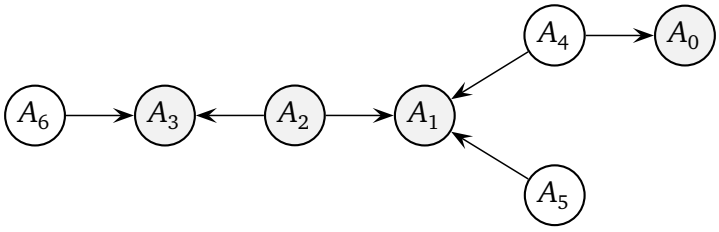


Figure 6.3: The s-network $\mathscr{G}_I$ returned by the application of Algorithm 1 to a CCURD instance I on the Bayesian network of Figure 6.2. The s-nodes are displayed in gray.

c	a_2	a_6	$P(A_3 = \mathrm{T}\|c, a_2, a_6)$
c'	T	T	0.6
c'	T	F	0.7
c'	F	T	0.2
c'	F	F	0.8
c''	T	T	0.2
c''	T	F	0.9
c''	F	T	0.2
c''	F	F	0.4

Table 6.5: Conditional mass functions for node A_3.

Theorem 12 is the basis to solve also a class of CCUR problems. Let us therefore consider a generic classification problem with missing data, whose set of classes is $\mathscr{C} := \{c_1, \ldots, c_r\}$. For each pair of classes, we can consider the corresponding binary CCUR instance. For each binary CCUR instance, we consider two CCURD instances as follows. If the binary CCUR instance requires to compare the classes between c_i and c_j, the first CCURD instance checks whether or not $c_i > c_j$, while the second checks $c_j > c_i$. Whenever one of these CCURD instances is true, the dominated class is rejected. The following algorithm reports the full procedure detecting the optimal classes:

Algorithm 2. *The procedure to solve a CCUR instance with set of classes* $\mathscr{C} := (c_1, \ldots, c_r)$*. The output is the set of the optimal classes* $\mathscr{C}_{opt}$*.*

```
1 C_opt := C;
2 for i = 1,...,r {
3     for j = 1,...,r {
4       if i < j {
5         if c_i>c_j { remove c_j from C_opt;}
6         if c_j>c_i { remove c_i from C_opt;}}}}
7 return C_opt;
```

Concerning the computational complexity of Algorithm 2, the total number of solved CCURD instances is quadratic in the input size, being exactly $r \cdot (r-1)$.

6.5 Solving Problems on S-Networks

In this section we show that the minimum of an s-network can be regarded as a full marginal of a joint valuation in a VA, as defined in Section 6.2.3. Further-

more, the fusion algorithm (6.5) can be employed to minimize s-networks. In the special case of s-polytrees, the algorithm takes linear time for an appropriate elimination sequence.

6.5.1 Minima of S-Networks as Local Computations on Valuation Algebras

We firstly introduce the following:

Theorem 13. *Let $\mathcal{G}$ be an s-network. Let $\mathcal{V}$ be the nodes of $\mathcal{G}$ and Φ the set of all the nonnegative real functions of any possible subset of $\mathcal{V}$. Let d be the map returning the variables in the argument of those functions and $\otimes$ the pointwise function product. Let also* El *be a variable elimination defined as $\phi^{-A_i} := \min_{a_i \in \mathcal{A}_i} \phi$. Thus, $(\Phi, \mathcal{V}, d, \otimes, \mathrm{El})$ is a valuation algebra and the potentials of $\mathcal{G}$, say $\{\phi_i\}_{i=0}^m$, are valuations in Φ.*

Proof. It is obvious to see that the operations of labelling, combination and variable elimination defined as in the statement of Theorem 13 are well defined according to the definition of VA in Section 6.2.3. In order to prove the theorem, it is therefore sufficient to check that the five axioms are satisfied. The commutativity and associativity of $\otimes$ naturally comes from the same property satisfied by the pointwise product between function and (A1) is therefore satisfied. It is also obvious to observe that the argument of a product of two functions is the union of the arguments of the functions and therefore also (A2) holds. The argument of $\phi^{-A_i} = \min_{a_i \in \mathcal{A}_i} \phi$ is clearly the argument of ϕ deprived by A_i. Thus, also (A3) is satisfied. Regarding (A4), the minimization of a function over two variables on its arguments are independent from the order of minimization and also this axiom holds. if ψ and ϕ are two valuations and A_i is only in the argument of ψ, then the minimization over A_i of the product of this two functions is the product between ϕ and the minimum of ψ. That means $(\psi \otimes \phi)^{-A_i} = \psi \otimes \phi^{-A_i}$, i.e., also (A5) holds. □

Accordingly, we can express the minimum of $\mathcal{G}$ as follows:

Theorem 14. *Let $\mathcal{G}$ be an s-network, $\{\phi_i\}_{i=0}^m$ its potentials and $(\Phi, \mathcal{V}, d, \otimes, \mathrm{El})$ the corresponding VA as in Theorem 13. Then,* $\min \mathcal{G} = (\phi_0 \otimes \cdots \otimes \phi_m)^{-\mathcal{V}}$.

Proof. It is sufficient to rewrite the products between potentials in (6.10) as combinations, according to the definition in the statement of Theorem 13, and the minimization as a (full) variable elimination. □

The fusion algorithm can therefore be used to calculate the minimum of an s-network. In general, the computation takes exponential time. Nevertheless, for a particular topology of the s-network $\mathscr{G}$, and an appropriate choice of the ordering in which the variables of $\mathscr{V}$ are eliminated, the algorithm becomes efficient. That is shown in the following.

6.5.2 Nodes Sorting on S-Polytrees

We call *s-polytree* an s-network $\mathscr{G}$ such that the underlying graph is a polytree. As an example, the s-network in Figure 6.3 is an s-polytree. The set $\mathscr{V}$ of the nodes of an s-polytree $\mathscr{G}$ has a natural structure of metric space. Given two nodes U and V, there is a single undirected path connecting them. Let $\delta(U,V)$ be the number of edges making up this path. The map δ is clearly a metric over $\mathscr{V}$ and $\delta(U,V)$ is called the *distance* between U and V. Let us call *neighbors* of U the nodes of $\mathscr{V}$ at distance one from U.

Given an s-polytree $\mathscr{G}$, an s-node A_k of $\mathscr{G}$ is called *lonely* if there is a node U of $\mathscr{G}$ such that A_k is the s-node at maximum distance from U (or one of them, if there are many). As an example, in the s-polytree of Figure 6.3, A_0 is the s-node at maximum distance from A_6 and can therefore be regarded as a lonely node. The lonely nodes of an s-polytree can be characterized as follows:

Theorem 15. *Let $\mathscr{G}$ be an s-polytree with at least two s-nodes and A_k a lonely node of $\mathscr{G}$. The variables in A_k^+, with the exception of a single variable S, appear only in the argument of ϕ_k.*

In order to prove Theorem 15, we first need the following result.

Lemma 4. *Let $\mathscr{G}$ be an s-polytree with at least two s-nodes. Let A_k be a lonely node of $\mathscr{G}$. Then the following holds:*

(i) *A_k has at most a special neighbor.*

(ii) *A_k can have special siblings, but all these siblings have a single parent in common with A_k, that is the same for all of them.*

(iii) *If A_k has actually a special neighbor A_l, the possible special siblings of A_k should have in A_l the single parent in common with A_k.*

Proof. Let U be a node of $\mathscr{G}$ such that A_k is the s-node of $\mathscr{G}$ (or one of them, if there are many) at maximum distance from U. The undirected path from U to A_k is unequivocally determined, because $\mathscr{G}$ is a polytree. Clearly $U \neq A_k$, because otherwise another s-node of $\mathscr{G}$ would be more distant from U than A_k.

Therefore, the path includes at least two nodes. Let S be the node preceding A_k in the path.

With the only possible exception of S, the neighbors of A_k cannot be special. If A would be a special neighbor of A_k different from S, the undirected path between U and A would cross A_k and A would be an s-node more distant from U than A_k. That proves (i).

If A_k has special siblings, all these s-nodes should have S as common parent. If A would be a special sibling of A_k through a common parent different from S, A would be an s-node more distant from U than A_k. That proves that S is the only parent common to A_k and its special siblings, as stated by (ii).

Finally, it was already proved that, if A_k has a special neighbor, this is S. Therefore the parent common to the special siblings of A_k, if actually A_k has a special neighbor, is exactly this neighbor. That proves (iii). □

Proof of Theorem 15. Let us first consider the case where A_k has not special neighbors. According to Lemma 3, A_k^+ can share some variables only with the vector variables corresponding to the possible special siblings of A_k. As a consequence of (ii) in Lemma 4, all the siblings of A_k have a single parent in common with A_k, that is the same for all of them. Let S be this node. S is clearly the only variable of A_k^+ that can appear also in some other vector variable.

Otherwise, if A_k has some special neighbor, then this is unique because of (i) in Lemma 4. Let A_l be this s-node. Point (iii) in Lemma 4 states that, if A_k has some special sibling, A_l should be a parent of A_k and also parent of all these siblings. Therefore, if A_l is child of A_k, A_k cannot have special siblings. In this case, A_k^+ can share some variable only with A_l^+ and, clearly, the only shared variable is A_k.

Finally, if A_l is parent of A_k, A_k can have some special sibling. We have already observed that A_l should be parent of all these siblings. Lemma 3 states that A_k^+ can share its variables only with A_l^+ and with the vector variables associated to the possible special siblings of A_k. In any case, A_l is the only variable of A_k^+ appearing also in some other vector variable. □

As an example, in the case of the s-polytree of Figure 6.3, $A_0^+ = (A_0, A_4)$, and while A_0 appears only in the argument of ϕ_0, A_4 appears also in ϕ_1.

Given a lonely node A_k, we denote by $\tilde{A}_k^+$ the vector variable that includes all the variables in A_k^+ except S and we refer to these variables as the *extreme leaves* of the s-polytree $\mathcal{G}$ with respect to A_k. In the case of s-polytrees with a single s-node, all the nodes are extreme leaves.

In the example of Figure 6.3, A_0 is the only extreme leave of $\mathcal{G}$ with respect to A_0 itself.

An s-node A_l is called *conjugate* node of a lonely node A_k, if the variable $S \in A_k^+$, which is not included in $\tilde{A}_k^+$, appears also in A_l^+. We can therefore regard S as the intersection of A_k^+ and A_l^+.

For example, A_1 is clearly the only conjugate of A_0 in the s-polytree of Figure 6.3, and A_4 can be regarded as the intersection between $A_0^+ = (A_0, A_4)$ and $A_1^+ = (A_1, A_2, A_4, A_5)$.

Call *siblings* two distinct children of the same parent. The conjugate nodes of a lonely node are characterized by the following:

Theorem 16. *Let A_k be a lonely node of an s-polytree $\mathcal{G}$ with at least two s-nodes. The conjugate nodes of A_k are the special neighbors and the siblings of A_k. Furthermore, A_k has at most a special neighbor; and if no s-nodes lie in the neighborhood of A_k, then A_k has at least one sibling.*

Proof. All the special neighbors and the special siblings of A_k are conjugate nodes of A_k because of Lemma 3. On the other side, if A_k is a lonely node of $\mathcal{G}$ and A_l a conjugate node of A_k, then A_k^+ and A_l^+ should share some variable. Thus, always because of Lemma 3, A_k and A_l are neighbors or siblings.

Furthermore, A_k has at most a special neighbor because of (i) in Lemma 4.

If the neighbors of A_k are all non-special, they all should be parents of A_k. The reason is that the arcs of an s-network cannot terminate on a non-special node. For the same reason those non-special parents of A_k cannot have any parent. Nevertheless, at least one of them should have a child, because otherwise $\mathcal{G}$ would include only a single s-node. This child is a second endpoint of an arc of an s-network and it is therefore a special node. Let A_l be this s-node. Clearly, A_l is a special sibling of A_k. That proves that a lonely node with no special neighbors should have at least one special sibling. □

In the case of the s-polytree of Figure 6.3, A_0 has no special neighbor and its unique special sibling A_1 is the only conjugate of A_0.

As a consequence of Theorem 16, for each lonely node A_k, there is at least a conjugate A_l.

It is indeed possible to show that the *pruning* of the extreme leaves of $\mathcal{G}$ preserves the s-polytree structure, as stated in the following:

Theorem 17. *Let $\mathcal{G}$ be an s-polytree with at least two special nodes, and A_k a lonely node of $\mathcal{G}$. If A_k is marked as not special, the nodes in $\tilde{A}_k^+$ are removed from $\mathcal{G}$, and ϕ_k is dropped from $\{\phi_i\}_{i=0}^m$, then a new s-polytree $\mathcal{G}'$ is obtained.*

In order to prove Theorem 17, we first need the following result.

Lemma 5. *Let $\mathcal{G}$ be an s-polytree with at least two s-nodes. Let A_k be a lonely node of $\mathcal{G}$. If A_k has a special neighbor, the non-special parents of A_k are leaf nodes of the*

undirected tree obtained from $\mathcal{G}$ by dropping the orientations. The same holds also if A_k has not special neighbors, with the only exception of the non-special parent of A_k that is also parent of the special siblings of A_k.

Proof. According to Definition 1, the non-special nodes of $\mathcal{G}$ cannot receive incoming arcs. Thus, a non-special parent V of A_k is a leaf node of the undirected tree corresponding to $\mathcal{G}$ if and only if V has not any child in addition to A_k.

Let U be the node of $\mathcal{G}$ such that A_k is the s-node of $\mathcal{G}$ (or one of them, if there are many) at maximum distance from U.

If A_k has a special neighbor, it should be unique because of (i) in Lemma 4. Let A_l be this node. The undirected path from U to A_k should cross A_l, because otherwise A_l would be more distant from U than A_k. If a non-special parent of A_k would have a child, this node would be special by definition of s-network and would be an s-node more distant from U than A_k. This is against the definition of U. Thus, in this case, the non-special parents of A_k cannot have any child. That proves the first part of the Lemma.

If A_k has no special neighbors, it should have at least one special sibling because of Theorem 16. Point (ii) in Lemma 4 states that A_k and its special siblings have a single common parent, say S. The path from U to A_k crosses S, because otherwise the special siblings of A_k would be more distant from U, than A_k. Thus, the non-special parents of A_k different from S cannot have any child, because otherwise their child would be s-nodes more distant from U than A_k. That proves the second part of the lemma. □

Proof of Theorem 17. The nodes of $\tilde{A}_k^+$, removed from $\mathcal{G}$ to obtain $\mathcal{G}'$, appear only in the potential ϕ_k by definition of $\tilde{A}_k^+$. All the potentials associated to the s-nodes of $\mathcal{G}'$ are therefore well defined.

Let S be the variable of A_k^+ not included in $\tilde{A}_k^+$. If $S = A_k$, then $\tilde{A}_k^+$ includes all the parents of A_k, while, if $S \in \Pi_{A_k}$, $\tilde{A}_k^+$ should include A_k. In the first case, to obtain $\mathcal{G}'$, we remove from $\mathcal{G}$ all the arcs having A_k as second endpoint, while in the second case A_k itself is removed. In any case, the condition about the second endpoints of the arcs of an s-network is always satisfied by $\mathcal{G}'$. That proves that $\mathcal{G}'$ is an s-network.

In order to prove that $\mathcal{G}'$ is an s-polytree, let U be the node of $\mathcal{G}$ such that A_k is the s-node of $\mathcal{G}$ (or one of them, if there are many) at maximum distance from U.

If A_k actually has a special neighbor, say A_l, we distinguish whether A_k is a parent or a child of A_l.

If A_k is a parent of A_l, then A_k appears both in A_k^+ and A_l^+ and $\tilde{A}_k^+ = \Pi_{A_k}$. According to (i) in Lemma 4, A_l is the only special neighbor of A_k and all the parents of A_k should be non-special.

Therefore, to obtain $\mathcal{G}'$ from $\mathcal{G}$, we remove the non-special parents of the lonely node A_k. According to Lemma 5, these nodes are leaf nodes in the tree corresponding to $\mathcal{G}$. Thus, $\mathcal{G}'$ is a polytree.

If A_l is a parent of A_k, A_l appears both in A_k^+ and A_l^+. This means that $\tilde{A}_k^+$ is composed by A_k and the parents of A_k different from A_l.

The parents of A_k different from A_l cannot be special because of the point (i) in Lemma 4. These nodes are therefore non-special parents of a lonely node and they should be leaf nodes in the undirected tree corresponding to $\mathcal{G}$ because of Lemma 5.

Furthermore, A_k cannot have any child. The path from U to A_k crosses A_l because otherwise A_l would be more distant from U than A_k. If A_k would have a child, this node would be special because of Definition 1, resulting an s-node more distant from U than A_k.

To obtain $\mathcal{G}'$ from $\mathcal{G}$, we can therefore remove first the parents of A_k different from A_l. Once we have removed these leaf nodes, A_k has a single parent (namely A_l) and no children. Thus, removing also A_k, we obtain a polytree, that is $\mathcal{G}'$.

If A_k has not special neighbors, it should have at least a special sibling because of Theorem 16. All the special siblings of A_k have a single parent in common with A_k because of (ii) in Lemma 4. Let S be this non-special parent of A_k. $\tilde{A}_k^+$ includes A_k and the parents of A_k different from S.

According to Lemma 5, the parents of A_k different from S are leaf nodes in the undirected tree corresponding to $\mathcal{G}$.

A_k cannot have any child also in this case. The path from U to A_k crosses S because otherwise the special siblings of A_k would be more distant from U than A_k. If A_k would have a child, this node would be special because of Definition 1, resulting an s-node more distant from U than A_k.

It is therefore possible to obtain $\mathcal{G}'$ from $\mathcal{G}$, removing first the parents of A_k different from S. Once we have removed these leaf nodes, A_k has a single parent (namely S) and no children. Thus, removing also A_k, we obtain a polytree, that is exactly $\mathcal{G}'$. □

As an example, the s-polytree of Figure 6.3 becomes a new s-polytree with three s-nodes after the pruning of A_0 (and the removal of ϕ_0). A lonely node in a pruned s-polytree $\mathcal{G}'$ can be characterized by the following:

Theorem 18. *Let $\mathcal{G}$ be an s-polytree. Given an arbitrary node of $\mathcal{G}$, say U, let A_k and $A_{k'}$ be respectively the first and the second s-nodes at maximum distance from U (or one of them, if there are many). Let $\mathcal{G}'$ be the s-polytree obtained marking A_k as not special, removing the nodes in $\tilde{A}_k^+$ from $\mathcal{G}$, and ϕ_k from the set of potentials, as in Theorem 17. Thus, $A_{k'}$ is a lonely node of $\mathcal{G}'$.*

In order to prove Theorem 18, we first need the following results.

Lemma 6. *Let $\mathscr{G}$ be an s-polytree with at least two s-nodes. Let A_k be a lonely node of $\mathscr{G}$ and U a node of $\mathscr{G}$ such that A_k is the s-node of $\mathscr{G}$ (or one of them, if there are many) at maximum distance from U. Then, U cannot be included in $\tilde{A}_k^+$.*

Proof. Because of its definition, $\tilde{A}_k^+$ cannot include U, if $\delta(U,A_k) > 1$ and also if U is a child of A_k.

If U is a parent of A_k and it is also special, U should appear both in A_k^+ and U^+. Therefore, U cannot be included in $\tilde{A}_k^+$.

If U is a non-special parent of A_k, let A_l be a second s-node of $\mathscr{G}$. A_l should be a neighbor of U, because otherwise it would be more distant from U than A_k. According to Definition 1, A_l cannot be a parent of the non-special node U. Thus, A_l is a child of U. This means that U appears both in A_k^+ and A_l^+, and therefore cannot be in $\tilde{A}_k^+$.

Finally, it is obvious to see that, U cannot coincide with A_k, because otherwise the remaining s-nodes of $\mathscr{G}$ would be more distant from U than $A_k = U$. □

Lemma 7. *A_k is the only special node that can appear in $\tilde{A}_k^+$.*

Proof. A_k is clearly the only special node included in A_k^+ if A_k has no special neighbors. Thus, in this case, A_k is the only s-node that can be in $\tilde{A}_k^+$. If A_k has some special neighbor, then this is unique because of (i) in Lemma 4. Let A_l be this s-node. If A_k is a parent of A_l, A_k appears both in A_k^+ and A_l^+. This means that A_k cannot be in $\tilde{A}_k^+$. Thus, $\tilde{A}_k^+$ includes only the parents of A_k and none of them is special, because A_l is the only special neighbor of A_k. In this case, therefore, no s-nodes are in $\tilde{A}_k^+$.

Finally, if A_l is parent of A_k, all the parents of A_k different from A_l are non-special, because A_l is the only special neighbor of A_k. Thus, A_k and A_l are the only s-nodes of A_k^+. But A_l appears also in A_l^+ and cannot be in $\tilde{A}_k^+$. Thus, A_k is the only s-node that can appear in $\tilde{A}_k^+$. □

Proof of Theorem 18. U is not included in $\tilde{A}_k^+$ because of Lemma 6 and therefore it should be a node of $\mathscr{G}'$. Furthermore, all the s-nodes of $\mathscr{G}$ different from A_k are s-nodes of $\mathscr{G}'$ because of Lemma 7. Thus, $\mathscr{G}'$ includes U and all the s-nodes of $\mathscr{G}'$ except A_k. The removal of some arcs and some nodes from $\mathscr{G}$ to obtain $\mathscr{G}'$, which is connected because of Theorem 17, cannot modify the distances between U and the s-nodes different from A_k. That means that the $A_{k'}$ is the s-node of $\mathscr{G}'$ at maximum distance from U. □

In the case of the s-polytree $\mathscr{G}$ of Figure 6.3, A_1 is the second s-node, after A_0, at maximum distance from A_6 and can therefore be regarded as a lonely node of the pruned s-polytree $\mathscr{G}'$, obtained removing A_0 and ϕ_0 according to Theorem 17.

This pruning procedure described in Theorem 17 yields an s-polytree and can therefore be iterated until an s-polytree with a single s-node is returned. As a consequence of Theorem 18, if we sort the s-nodes of $\mathcal{G}$ according to their distance from U, the i-th element of this sequence is a lonely node of the s-polytree returned by the i-th iteration of the pruning procedure. For each node of this sequence, we can consider the corresponding extreme leaves. It is trivial to check that all the elements of $\mathcal{V}$ appear in this collection of extreme leaves.

The discussion in this section suggests the opportunity to employ this collection of extreme leaves as an elimination sequence for the variables in $\mathcal{V}$ in order to minimize s-polytrees through the fusion algorithm. It is finally clear that, in the case of s-polytrees with a single s-node, there is a single potential and therefore no particular strategy to detect an efficient elimination sequence is required.

6.5.3 Solution Algorithm

If $\mathcal{G}$ is an s-polytree and A_k is a lonely node of $\mathcal{G}$, the elimination of the extreme leaves of $\mathcal{G}$ with respect to A_k can be restricted to ϕ_k because of Axiom (A5). Thus:

$$(\phi_0 \otimes \ldots \otimes \phi_m)^{-\tilde{A}_k^+} = (\otimes_{i=0,\ldots,m/i\neq k}\phi_i) \otimes \phi_k^{-\tilde{A}_k^+}. \tag{6.15}$$

But $d(\phi_k^{-\tilde{A}_k^+})$, that is a single variable because of Theorem 15 and Axiom (A2), appears also in ϕ_l, where A_l is a conjugate of A_k, by definition of conjugate. With a simple redefinition of the potential of A_l:

$$\phi_l = \phi_l \otimes \phi_k^{-\tilde{A}_k^+}, \tag{6.16}$$

the information associated to the potential ϕ_k after the elimination of the variables in $\tilde{A}_k^+$ can be embedded in ϕ_l. Notably, $d(\phi_l') = d(\phi_l)$ because of Axiom (A2), that means that the potential redefinition in Equation (6.16) does not affect the domain of ϕ_l. Finally, if we drop the potential ϕ_k from the set of potentials, a new s-polytree $\mathcal{G}'$ is obtained because of Theorem 17 and the procedure can be iterated. The overall procedure, returning the minimum of the s-polytree because of Theorem 14, is reported in the following algorithm:

Algorithm 3. *The findMin routine. In input we have an s-polytree $\mathcal{G}$. The subroutine findInter returns a vector with the intersection of two arrays of variables.*

```
1 U := randomly chosen node of G;
2 (δ_0,...,δ_m) := findDistances(G,U);
3 while number of s-nodes in G > 1 {
```

```
4       k := arg max_j δ_j;
5       A_l := findConjugate(A_k,𝒢);
6       S := findInter(A_k^+,A_l^+);
7       Ã_k^+ := remove S from A_k^+ ;
8       φ_l = φ_l ⊗ φ_k^{-Ã_k^+};
9       mark A_k as not special;
10      drop the nodes in Ã_k^+ from 𝒢;
11      remove φ_k, from (φ_0,...,φ_m);
12      remove δ_k, from (δ_0,...,δ_m);}
13  return φ_l^{-d(φ_l)};
```

The distances between a randomly chosen node U and the s-nodes of $\mathscr{G}$ are initially computed (lines 1–2 of Algorithm 3). The routine *findDistances*($\mathscr{G}$,U), returning the distances between U and the s-nodes of $\mathscr{G}$, can be implemented through the well known *depth first search* (DFS) algorithm [Eve79] over the undirected graph obtained forgetting the orientation of the arcs of $\mathscr{G}$.

A lonely node A_k of $\mathscr{G}$ can therefore be detected as the s-node at maximum distance from U (line 4). Algorithm 4, detects a conjugate A_l of A_k (line 5) and the extreme leaves of $\mathscr{G}$ with respect to A_k are therefore obtained (line 6–7). These variables are indeed eliminated and the result is embedded on ϕ_l as in (6.16) (line 8). Finally (lines 9–12), $\mathscr{G}$ is transformed by the pruning procedure of Theorem 17 in a new s-polytree with an s-node fewer. The overall procedure is iterated (line 3) until an s-polytree with a single s-node, whose minimization is trivial (line 13), is returned.

Algorithm 4. *The findConjugate function. The inputs are the polytree $\mathscr{G}$ and a lonely node A_k. The output findConjugate($\mathscr{G}$,A_k) is a conjugate of A_k. The subroutine findNeighbors returns the neighbors of the node in its argument.*

```
1   for each V ∈ findNeighbors(A_k) {
2       if V is special {
3           A_l := V;
4           go to 8; }
5       else  {
6           if V has a children W {
7               A_l := W; }}}
8   return A_l;
```

As an example, Algorithm 3 can be used to calculate the minimum of the s-polytree of Figure 6.3. The distances between $U := A_6$ and the s-nodes of $\mathscr{G}$

are: $\delta_0 = 5$, $\delta_1 = 3$, $\delta_2 = 2$, $\delta_3 = 1$. Thus, A_0 is a lonely node of $\mathcal{G}$ and its sibling A_1 is a conjugate of it.

Clearly, $\tilde{A}_0^+ = A_0$ and the redefinition of ϕ_1 should be:

$$\phi_1(a_1, a_2, a_4, a_5) = \phi_1(a_1, a_2, a_4, a_5) \cdot \min_{a_0 \in \mathcal{A}_0} \phi_0(a_0, a_4). \tag{6.17}$$

Furthermore, A_0 is marked as not-special and dropped from $\mathcal{G}$, the potential ϕ_0 is removed from $\{\phi_i\}_{i=0}^3$, and similarly δ_0 is removed from $\{\delta_i\}_{i=0}^3$. After these operations, $\mathcal{G}$ is now an s-polytree with three s-nodes. A_1 is clearly its s-node at maximum distance from A_6 and it is therefore a lonely node of $\mathcal{G}$, while A_2 is a conjugate of it. The extreme leaves are A_1, A_4 and A_5 and the redefinition of ϕ_2 is:

$$\phi_2(a_2) = \phi_2(a_2) \cdot \min_{a_1 \in \mathcal{A}_1, a_4 \in \mathcal{A}_4, a_5 \in \mathcal{A}_5} \phi_1(a_1, a_2, a_4, a_5). \tag{6.18}$$

A further iteration of the procedure yields to:

$$\phi_3(a_2, a_3, a_6) = \phi_3(a_2, a_3, a_6) \cdot \phi_2(a_2), \tag{6.19}$$

and finally, we conclude that the minimum of the s-polytree $\mathcal{G}_I$ is:

$$\min_{a_3 \in \mathcal{A}_3, a_2 \in \mathcal{A}_2, a_6 \in \mathcal{A}_6} \phi_3(a_2, a_3, a_6) = \frac{2}{3}. \tag{6.20}$$

According to Theorem 12, the CCURD instance I associated to $\mathcal{G}_I$ is therefore false and c' does not dominate c''.

Now, let $\overline{I}$ be the CCURD instance involving the decision whether or not $c'' > c'$ with all the attribute variables missing. We can proceed in complete analogy with the procedure used to solve I. The numerical value of the minimum of $\mathcal{G}_{\overline{I}}$ is $\frac{4}{189}$. $\overline{I}$ is therefore false and we conclude that the two classes are mutually undominated. Therefore, if all the attribute variables are missing, we are not able to identify a single optimal class and both the values c' and c'' are plausible.

Finally, to detect whether or not Algorithm 3 can be used to solve a given CCURD instance I, it is sufficient to check if the graph $\mathcal{G}_I$ returned by Algorithm 1 is a polytree. The condition $|\mathcal{V}| = |\mathcal{E}| + 1$ for $\mathcal{G}_I(\mathcal{V}, \mathcal{E})$ can therefore be used as an obvious applicability check. Note that Algorithm 1 obtains $\mathcal{G}_I$ removing some nodes and arcs from B^+. Therefore $\mathcal{G}_I$ can be a polytree also if the original Markov blanket is multiply connected (e.g., the net in Figure 6.2).

Remember that we are focusing on connected s-networks. In the general case of a disconnected s-network $\mathcal{G}$, we have only to check whether or not the graph is a polyforest. In the positive case, Algorithm 3 can be used to calculate the minima of the s-polytrees associated to the connected component of $\mathcal{G}$ with at least one s-node, while the overall minimum is just the product of these minima because of Theorem 11.

Finally, concerning the efficiency of the overall procedure:

Theorem 19. *Algorithm 3 has linear complexity.*

Proof. The subroutine *findDistances* is known to be linear in the number of arcs of $\mathcal{G}$ [Eve79]. On the other hand, *findConjugate*$(A_k,\mathcal{G})$ requires a number of operations equal to the number of neighbors of A_k. The children of A_k should be s-nodes because of Definition 1, while A_k has at most a special neighbor, and hence a children, because of (i) in Lemma 4. That means that A_k has at most a children. The number of neighbors of A_k is therefore dominated by $|\Pi_{A_k}|+1$. The subroutine *findConjugate* is invoked m times, and the overall number of operations can therefore be bounded by $\sum_{i=0}^{m}|\Pi_{A_k}|+m$. But the first term of this sum represents the number of arcs of $\mathcal{G}$ because of Definition 1. Thus, also this part of the algorithm takes only a linear number of operations. Finally the evaluation of $\phi_k^{-\tilde{A}_k^+}$ was already noted to take place in a domain of the same dimension of those defined in input. That proves the thesis. □

Note that in analogy with [dCZ04, Section 6], the common technique called *loop cutset conditioning* can be used to solve a CCUR instance I even if the graph $\mathcal{G}_I$ returned by Algorithm 1 is not a polyforest. In this case the computation takes exponential time.

6.6 Notes on CUR-Based Classification

So far we have focused on algorithms for CUR-based classification with Bayesian networks. In this section, we would like to give a broader perspective of this approach so as to clarify its characteristics and possible usages.

An important point concerns the cautiousness of CUR. Remember that CUR assumes near-ignorance about the missingness process, and this implies having to consider all the completions of the missing values as part of the updating rule. Not surprisingly, this procedure is likely to yield partially indeterminate conclusions (i.e., classifications), especially when there are missing attribute variables that are important to predict the class. Avoiding indeterminacy is therefore tightly connected with being able to measure all good predictors. This will probably not be the case at the initial stages of interaction with an expert system, in which only some of the variables are measured. But the interaction is often a dynamic rather than a static process (this is very natural with credal classifiers, and more generally with imprecise probability models) in which more and more measures are collected along the way towards definite conclusions. This dynamic way of using expert systems would eventually lead CUR to yield strong enough conclusions, with the advantage of having the intermediate conclusions, guiding the process, not biased by potentially strong assumptions about the missingness process.

Obtaining stronger conclusions would be favored also by modifying CUR in such a way that it may apply to incomplete rather than only to missing observations. An incomplete observation is defined as a set-based observation that does not necessarily coincide with the entire possibility space (as with missing data); the fact that some values may be excluded obviously favors obtaining stronger conclusions. It is worth pointing out that the algorithms presented in this chapter for CUR can be immediately extended to incomplete observations of attributes: whenever there is a minimization, it is sufficient for the extension to minimize over the observed subset an attribute's possibility space rather than over the entire space. It should also be noted that the evolution of CUR towards incomplete observations has already been proposed under the name of *conservative inference rule* [Zaf05], which actually extends CUR under more substantial respects, for instance by establishing the theoretical underpinning for statical applications of these conservative rules.

Given that the last observation points to possible uses of CUR in a statistical pattern classification context, it may be useful to briefly discuss the topic. One important thing to be aware of is that rules such as CUR find justification in a statistical classification setting that produces (complete) data in an independently and identically distributed way, when the missingness process is *not* identically distributed, i.e., when each unit of (complete) data may be subject to a different missingness process.[4] Interestingly, in such a setup traditional precise (i.e., non-credal) classifiers cannot be really considered competitors of credal classifiers when the missingness processes is (partly) unknown: it is very easy to build applications that make every precise classifier fail to predict the right classes; and this may be even done so that there is no way to know such bad performance in advance by making the empirical tests traditionally employed in the classification practice (see [Zaf05, Section 6]). The considered setting seems therefore particularly suited for CUR-based classification and its extensions, and worth exploring.

6.7 Summary and Conclusions

Probabilistic expert systems suggest actions on the basis of the available evidence about a domain. Often such an evidence is only partial, due to a number of reasons such as economic or time constraints. In order for the suggested actions to be credible, it is important to properly take into account the process that makes the evidence partial by hiding the state of some of the variables used to describe the domain. The recently derived conservative updating rule achieves this by

[4] In fact, if the missingness process is also (independently and) identically distributed, a more traditional approach should be employed [Zaf05, Section 5].

considering a near-ignorance about the missingness process, and by updating beliefs accordingly. In order to make the rule profitably used it is important to develop efficient algorithms to compute with it.

In this chapter we have shown that it is not possible in general to create efficient algorithms for such a purpose (unless P=NP): in fact, using the conservative updating rule to do efficient classification with Bayesian networks is shown to be NP-hard. This parallels analogous results with more traditional ways to do classification with Bayesian nets: in those cases, the computation is efficient only on polyforest-shaped Bayesian networks. Our second contribution shows that something similar happens using the conservative updating, too. Indeed we provide a new algorithm for robust classification that is efficient on polyforest-shaped s-networks. This extends substantially a previously existing algorithm which, loosely speaking, is efficient only on disconnected s-networks.

Yet, it is important to stress that the computational difference between traditional classification with Bayesian nets and robust classification based on the conservative updating rule is remarkable: first, the former is based on the entire net, while the latter only on the net made by the class variable with its Markov blanket; second, while the former needs that the entire network is a polyforest in order to obtain efficient computation, the latter requires only that the associated s-network is. This means that the computation is efficient also in many cases when the class variable with its Markov blanket forms a multiply connected net in the original Bayesian network. In other words, computing robust classifications with the conservative updating will be typically much faster than computing classifications with the traditional updating rule. Given that the latter classifications are necessarily included in the former, by definition of the conservative updating rule, it seems to be worth considering robust classifications not only as a stand-alone task, but also as a pre-processing step of traditional classification with Bayesian nets.

With respect to future research, a natural development would be a generalization of our algorithms to the conservative inference rule (see Section 3.1.1), which models also an hybrid situation of near-ignorance about missingness process of some variables and MAR condition satisfied by the others. It seems also possible to proceed as in [dCZ04, Section 7] to employ our algorithm also in the case of *credal networks* [Coz00].

Chapter 7

Credal Networks for Military Identification Problems

In this chapter, we present a credal network for risk evaluation in case of intrusion of civil aircrafts into a no-fly zone. The different factors relevant for this evaluation, together with an independence structure over them, are initially identified. These factors are observed by sensors, whose reliabilities can be affected by variable external factors, and even by the behavior of the intruder. A model of these observation mechanisms, and the necessary fusion scheme for the information returned by the sensors measuring the same factor, are both completely embedded into the structure of the credal network according to the formalism developed in the first part of this thesis. A pool of experts, facilitated in their task by specific techniques to convert qualitative judgments into imprecise probabilistic assessments, has made possible the quantification of the network. We show the capabilities of the proposed network by means of some preliminary tests referred to simulated scenarios. Overall, we can regard this application as an useful tool to support military experts in their decision, but also as a quite general imprecise-probability paradigm for information fusion.[1]

7.1 Protection of No-Fly Areas

In the recent times, the establishment of a restricted or prohibited flight area around important potential targets surveyed by the Armed Forces has become usual practice, also in neutral states like Switzerland, because of the potential danger of terror threats coming from the sky. A *prohibited flight area* is an airspace of definite dimensions within which the flight of aircraft is prohibited. A

[1]The work presented in this chapter has been done in cooperation with Alberto Piatti and Ralph Brühlmann.

restricted flight area is an airspace of definite dimensions within which the flight of aircrafts is restricted in accordance with certain specified conditions [Ser07]. In particular we refer to the Swiss case, where restricted flight areas are usually established to protect international conferences (e.g., World Economic Forum in Davos).

Once a restricted flight area is issued for the protection of a single strategic object, all the aircrafts flying in this region without the required permissions are considered *intruders*. The restricted flight area can be imagined as divided in two concentric regions: an external area, devoted to the identification of the intruder, where the intruder is observed by many sensors of the civil and military *Air Traffic Control*, and an internal area, called *killing box*, which is a small region containing the object to protect and the military units, where fire is eventually released if the intruder is presumed to have bad aims.

Clearly, not all the intruders have the same intentions: there are intruders with bad aims, called *renegades*, intruders with provocative aims, erroneous intruders, and even aircrafts that are incurring an emergency situation. Since only renegades represent a danger for the protected object, the recognition of the intruder's aim plays a crucial role in the following decision, which, if it is wrong, is going to be critical. This is the identification problem we address in this chapter.

The problem is complex for many reasons: (i) the risk evaluation usually relies on qualitative expert judgments; (ii) it requires the fusion of information coming from different sensors, and this information can be incomplete or partially contradictory; (iii) different sensors can have different levels of reliability, and the reliability of each sensor can be affected by exogenous factors, as geographical and meteorological conditions, and also by the behavior of the intruder. A short review of the problem and some detail about these difficulties is reported in Section 7.2.

We regard credal networks as the appropriate mathematical paradigm for the modeling of military identification problems, as they are particularly suited for modeling and doing inference with qualitative, incomplete, and also conflicting information.

More specifically, we have developed a credal network for the considered identification problem. This is achieved by a number of sequential steps: determination of the factors relevant for the risk evaluation and identification of a causal structure between them (Section 7.3.1); quantification of this qualitative structure by imprecise probabilistic assessments (Section 7.4.1); determination of a qualitative model of the observation mechanism associated to each sensor, together with the necessary *fusion scheme* of the information collected by the different sensors (Section 7.3.2); quantification of this model by probability intervals (Section 7.4.2). An analysis of the main features of our imprecise-

probability approach to information fusion is indeed reported in Section 7.5.

The credal network is finally used to evaluate the level of risk, which is simply the probability of the risk factor conditional on the information collected by the sensors in the given scenario. A description of the approximate procedure used to update the network, together with the results of some simulations, is reported in Section 7.6.

Summarizing, we can regard this model as a practical tool to support military experts in their decisions for this particular problem. But, at the same time, this credal network can be regarded as a prototypical modeling framework for general identification problems requiring information fusion.

7.2 Military Aspects

This section focuses on the main military aspects of the identification problem addressed by this chapter. Let us first report the four possible values of the *risk factor* by which we model the possible intentions of the intruder.

(i) *Renegade*: the intruder intends to use itself as a weapon to damage the strategic target defended by the restricted flight area.

(ii) *Agent provocateur*: the aim of an agent provocateur is to provoke or demonstrate. An agent provocateur knows exactly what it is doing and does not want to die, therefore it is expected to react positively at a certain moment to radio communications.

(iii) *Erroneous intruder*: the intruder is an aircraft entering the restricted flight zone because of an error in the flight path due to bad preparation of the flight or to bad level of training of the pilot.

(iv) *Damaged intruder*: a damaged intruder is an aircraft without bad aims that is incurring an emergency situation due to a technical problem. The pilot does not necessarily know what he is doing because of a possible situation of panic. A damaged intruder can react negatively to radio communications, as their instruments could be switched off because of electrical failures. A proper identification of damaged intruders is very important because they can be easily confused with renegades.

In order to decide which one among these four categories reflects the real aim of the intruder an appropriate *identification device* should be set up. Figure 7.1 displays a typical structure for the identification devices employed in Switzerland. When a restricted flight area is set up for the protection of an important object, the *Air Defence Direction Center* (ADDC) is in charge of the identification

of possible intruders. The ADDC collects the information provided by three main sources: (i) the sensors of the civil *Air Traffic Control* (ATC), (ii) the sensors of the military ATC, (iii) the interceptors of the Swiss Air Force devoted to Air Police missions. Once this evidential information has been collected, the ADDC performs the identification of the aim of the intruder.

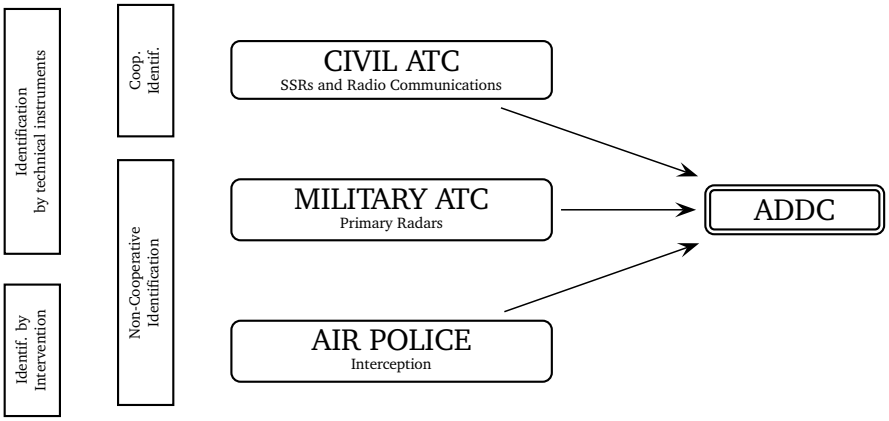


Figure 7.1: The structure of the identification device.

The civil ATC sensors are based on a collaborative communication between the ATC and the intruder. In fact, the detection of the intruder by the ATC is possible only if the intruder is equipped and operates with a *transponder*. Transponders are electronic devices that, if interrogated by the civil ATC radar, emit a signal enabling a two-dimensional localization of the aircraft. Radars based on this principle are called *Secondary Surveillance Radars* (SSRs). Transponders also report data about the height of the aircraft, as measured by their instruments, allowing in fact for a tridimensional positioning. Transponders emit also an *identification code*. In our model, we consider the identification code *Mode 3/A*, that, in certain cases, does not allow the exact identification of the intruder to be realized (e.g., all the aircrafts flying according to the *visual flight rules* emit the same code).[2] It should be also pointed out that, if the transponder is switched off, the intruder remains invisible to the civil ATC because the SSR is unable to detect it.

Overall, we summarize the information relevant for the identification gathered by the civil ATC in terms of two distinct factors: the TRANSPONDER MODE

[2]The most informative identification code *Mode S* is not considered here, because it has not yet been implemented extensively in practice.

3/A,[3] indicating if and eventually what type of identification code has been detected by the SSR; the ATC REACTION, describing the reaction of the intruder to the instructions that civil ATCs, reported to intruders flying in the direction of the restricted area, in order to deviate them from their current flight route.

Unlike civil ATC sensors, sensors managed by the military ATC and military Air Police units are based on a non-collaborative observation of the intruder. The main military ATC sensors are the *Military Radar Stations*, detecting echoes reflected by the intruder of radio pulses emitted by the radar. These radars provide in a continuous way the tridimensional position of the intruder. The other military sensors, that are particularly suited for the identification of intruders flying at relatively low height, are the pointing devices of anti-air firing units (two-dimensional and tracking radars, TV and infrared cameras) and the *Ground Observer Corps* (GOC), which are military units equipped with optical instruments to observe the intruder from the ground.

The information gathered by these sensors which is relevant for the identification of the intentions of the intruder can be summarized by the following factors: AIRCRAFT HEIGHT, HEIGHT CHANGES, ABSOLUTE SPEED, FLIGHT PATH, AIRCRAFT TYPE, and also REACTION TO ADDC, which is the analogous of REACTION TO ATC, but referred to the case of detection (and communication) by the military ATC.

Finally, regarding the information gathered by the interceptors of the Swiss Air Force, which is reported to the ADDC, the possible identification missions of the interceptors are divided into three categories according to the International Civil Aviation Organization: *surveillance*, *identification* and *intervention*. In the first type of mission, the interceptor does not establish a visual contact with the intruder but observes its behavior using sensors,[4] in this case the interceptor is considered as a sensor observing the same factors as the other sensors of the civil and military ATC. In the second and in the third type of missions, the interceptor establishes a visual contact with the intruder with the intention of observing it (identification), or giving it instructions in order to deviate the aircraft from the current flight route, or also to land it (intervention). The reaction of the intruder to interception is very informative about its intentions. We model this reaction to the latter two types of mission by the factor REACTION TO INTERCEPTION.

The intruder is assumed to be observed during a sufficiently long time window called *observation period*. All the factors we have defined to describe the behavior of the intruder during this observation period are discrete random vari-

[3]The following typographical convention is used: random variables to be considered as relevant factor for the identification are written in SMALL CAPITALS, while the possible states of these variables are written in *slanted lowercase*.

[4]E.g., the most important interceptor of the Swiss Air Force, the Boeing F/A 18, is equipped with a powerful board radar.

ables, whose possible values have been defined with respect to the dynamic component of the identification process, in order to eliminate the dependency of the model on local issues (e.g., geographical issues). To explain how these aspects are taken into account, we detail the definitions of the factors FLIGHT PATH and HEIGHT CHANGES. The first factor describes the route followed by the intruder during the observation period from an intentional point of view and not from a physical or geographical perspective. Accordingly, their possible values are defined as follows.

(i) *Suspicious Route*: the intruder follows a suspicious flight route in direction of the protected objects.

(ii) *Provocative Route*: the intruder flights in the restricted area without approaching significantly the protected objects in an apparently planned way.

(iii) *Positive Reaction Route*: the intruder corrects its flying route spontaneously or according to instructions of the ATC or interceptors.

(iv) *Chaotic Route*: the intruder follows an apparently chaotic flight path.

The definition of these possible states is independent of the specific geographical situation. In practice, we assume that a route observed on the radar or on other sensors by the ADDC is interpreted from an intentional point of view in the light of the current geographic situation. Similarly, the factor HEIGHT CHANGES describes the behavior of the intruder with respect to its altitude, by the following possible values.

(i) *Climb*: the intruder is climbing, i.e., increasing its altitude.

(ii) *Descent*: the intruder is descending. i.e., decreasing its altitude.

(iii) *Stationary*: the intruder maintains roughly the same altitude during the observation period.

(iv) *Unstable*: the intruder climbs and descends in an alternate way.

These values reflect an observation of the dynamic behavior of the intruder during the whole observation period.

Another important issue regarding our model is the description of the sensors available in the identification device and the evaluation of their efficiency. Consider, for instance, a situation where a number of GOCs are observing the FLIGHT PATH of an intruder. Assuming this number low, the corresponding observation is probably of low quality, due to the scarce *presence* of GOCs. Now consider a restricted flight area completely surveyed by GOCs, where the meteorological

condition is characterized by continuous low clouds. Despite the high presence of GOCs, also in this case the observation is probably of low quality. In this case, the reason is the scarce *reliability* of GOCs.

By these two examples, we intend to point out that a proper description of the identification device can be obtained by distinguishing between the presence and the reliability of each sensor. The presence depends on the specific architecture of the identification device, on the technical limits of the sensors, and also on the behavior of the intruder itself, being in particular affected by the AIRCRAFT HEIGHT (e.g., some sensors can observe the intruder only if it is flying at low heights). The *reliability* depends on the meteorological and geographical conditions, on specific technical limits of the sensors (e.g., radars have low quality in the identification of the AIRCRAFT TYPE, independently of its presence) and on the AIRCRAFT HEIGHT. All these aspects are implicitly considered by the Expert that is required to specify directly the presence and the reliability of the different sensors. This model of the identification device is detailed in Sections 7.3 and 7.4.

7.3 Qualitative Assessment of the Network

We are now in the position to describe the credal network developed for our application. According to the discussion in the previous section, this task first requires the qualitative identification of the conditional dependencies between the different variables involved in the model, which can be coded by a corresponding directed acyclic graph.

As detailed in Section 7.2, the variables we consider in our approach are: (i) the RISK FACTOR, (ii) the nine variables used to assess the intention of the intruder, (iii) the variables representing the observations returned by the sensors, (iv) for each observation two additional variables representing presence and reliability of the observation with the sensor. In the following, we refer to the variables in the categories (i) and (ii) as *core variables*.

7.3.1 Risk Evaluation

Figure 7.2 depicts the conditional dependencies between the core variables according to the military and technical considerations of the Expert.[5] As an example, the arcs connecting the nodes AIRCRAFT TYPE, AIRCRAFT HEIGHT, and RISK FACTOR with the ABSOLUTE SPEED, correspond to the following Expert's remarks:

[5] We briefly call *Expert* a pool of military experts from the Swiss Air Force, we have consulted during the development of the model.

there is a strong relation between the height above the ground and the corresponding speed of an aircraft (technical considerations); *a renegade is expected to fly as fast as possible* (military consideration); *an intruder flying with a light aircraft, because of the limited maximal speed of this type of aircrafts, would necessarily flight very slowly*. The specification of this part of the network has required a considerable amount of military and technical expertise that, due to confidentiality reasons, cannot be explained in more detail here.

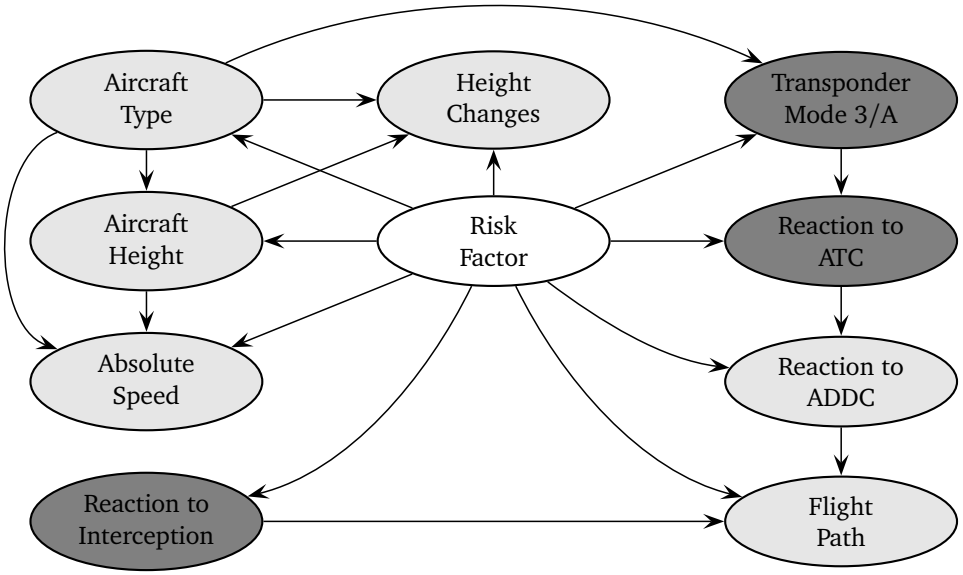


Figure 7.2: The core of the network. Dark gray nodes are observed by single sensors, while light gray nodes are observed by set of sensors for which the information fusion scheme in Section 7.3.2 is required.

7.3.2 Observation and Fusion Mechanism

We use the general definition of *latent* and *manifest variables* given by [SRH04]: a *latent variable* is a random variable whose realizations are unobservable (hidden), while a *manifest variable* is a random variable whose realizations can be directly observed. According to [BMvH02], there may be different interpretations of latent variables. In our model, we consider a latent variable as an unobservable random variable that exists independent of the observation. The *core variables*, in Figure 7.2, are regarded as latent variables that, to be determined, usually require the fusion of information coming from different sensors, with different levels of reliability. The observations of the different sensors are considered as manifest variables. Nevertheless, in the case of the identification code

emitted by the intruder (TRANSPONDER MODE 3/A), the REACTION TO INTERCEPTION observed by the pilot, and the REACTION TO ATC observed by the controllers through SSR, the observation mechanism is immediate; thus we simply identify the latent with the corresponding manifest variable, adding the value *missing*, as a further possible value of the variable.[6] This value is considered a possible value for every manifest variable and can have particular meanings (e.g., for the variable TRANSPONDER MODE 3/A the value *missing* probably means a switched off transponder).

Clearly, if the risk factor was the only latent variable, the network in Figure 7.2 would be the complete network needed to model the risk evaluation. But, because we are dealing with latent variables observed by many sensors, a model of the observation and a fusion mechanism has to be added to the current structure.

Observation Mechanism We begin by considering observations by single sensors, and then we explain the fusion scheme for several sensors. Consider the following example: suppose that an intruder is flying at low height and is observed by ground-based observation units in order to evaluate its FLIGHT PATH. For this evaluation, the intruder should be observed by many units. If our identification architecture is characterized by too a low number of observation units, it is probable that the observation of the flight path would be incomplete or even absent, although the meteorological and geographical conditions are optimal. In this case, the low quality of the observation is due to the scarce presence of the sensor. Suppose now that the architecture is characterized by a very large number of observation units but the weather is characterized by a complete cloud cover with low clouds, then the quality of the observation is very low although the presence of units is optimal. In this case the low quality of the observation is due to the low reliability of the sensor under this meteorological condition. This example motivates our choice to define two different factors affecting the quality of an observation by a single sensor: the RELIABILITY and the PRESENCE.

Figure 7.3 illustrates, in general, how the evidence provided by a sensor about a latent variable is assessed. The manifest variable depends on the relative *latent variable*, on the PRESENCE of the sensor and on its RELIABILITY. Both RELIABILITY and PRESENCE are categorical variables with three possible values, *high*, *medium* and *low* for the RELIABILITY, and *present*, *partially present* and *absent* for the PRESENCE.

[6] The manifest variables we consider are typically referred to the observations of corresponding latent variables. Thus, if X is a latent variable, the possibility space Ω_O of the corresponding manifest variable O takes values in the set Ω_X augmented by the supplementary possible value *missing* (we denote this value by '$*$').

According to the military principles outlined in Section 7.2, the RELIABILITY of a sensor can be affected by the meteorological and geographical situation and also by the AIRCRAFT HEIGHT, while, regarding the PRESENCE only the AIRCRAFT HEIGHT and architecture of the identification device affect the quality of the observations. The influence of the latent variable AIRCRAFT HEIGHT is related to the technical limits of the sensors: there are sensors that are specific of the low and very low heights, like tracking radars and TV or IR cameras; other sensors, like the 3D radars of the fixed military radar stations, are always present at high and very high heights, but are not always present at low and very low heights.

The meteorological and geographical conditions do not affect the PRESENCE of a sensor, but only its RELIABILITY. It is important to point out that these conditions are always observed and we do not display them explicitly as variables in the network, being already considered by the Expert during his quantification of the RELIABILITY.

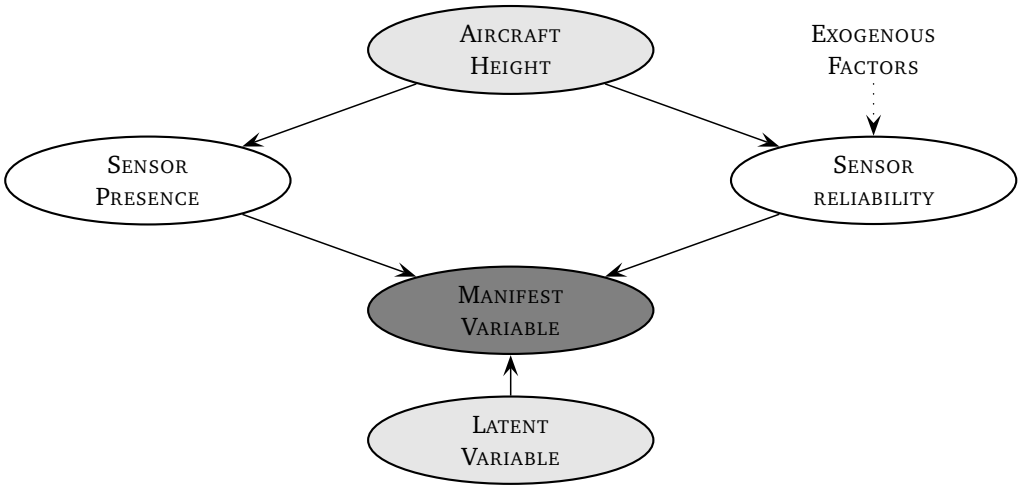


Figure 7.3: Observation mechanism for a single sensor. The *latent variable* is the variable to be observed by the sensor, while the *manifest variable* is the value returned by the sensor itself.

Sensors Fusion We can finally explain how the information collected by the different observations of a single latent variable returned by different sensors can be fused together. Consider, for example, the determination of the latent variable AIRCRAFT TYPE. This variable can be observed by four types of sensors:

TV cameras, IR cameras, ground-based observation units and air-based interceptors. For each possible sensor, we model the observation using a structure like the network in Figure 7.3: there is a node representing the PRESENCE of the sensor and a node representing the RELIABILITY, while the variable AIRCRAFT HEIGHT influences all these nodes. This structure permits the fusion of the evidence about the latent variables coming from the different sensors, taking into account the reliability of the different observations in a very natural way and without the need of any external specification of explicit fusion procedures. Section 7.5 reports a note on the main features of this approach, which has been inspired by similar techniques adopted for Bayesian networks [DO06].

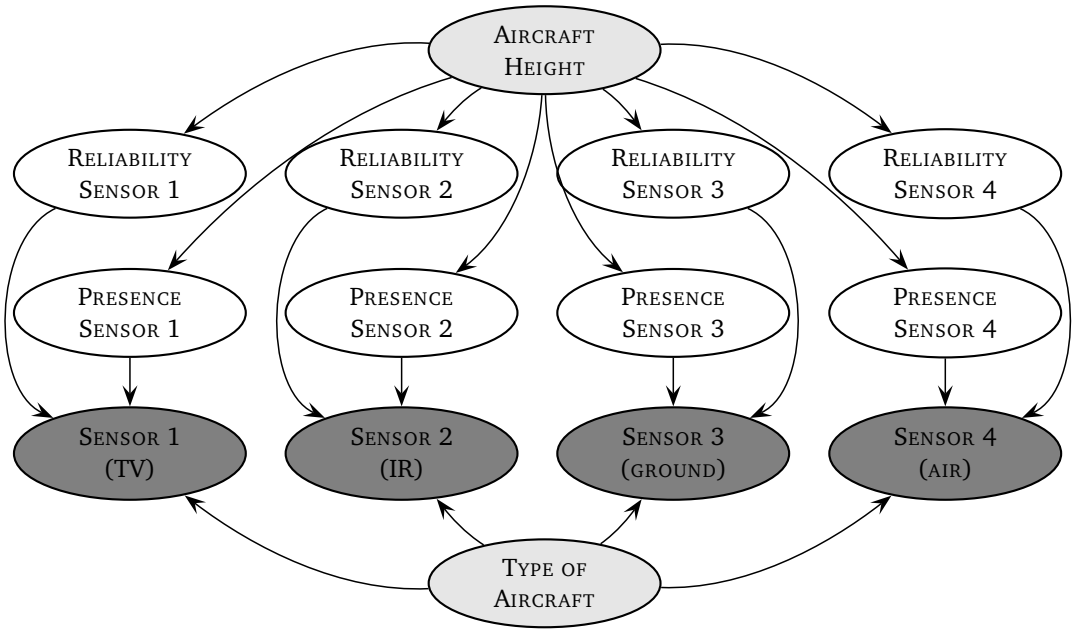


Figure 7.4: The determination of the latent variable TYPE OF AIRCRAFT by four sensors.

We similarly proceed for all the latent variables requiring the fusion of information from many sensors. This practically means that we add a subnetwork similar to the one reported in Figure 7.4 to each light gray node of the core network in Figure 7.2. The resulting directed graph, which is still acyclic, is shown in Figure 7.5.

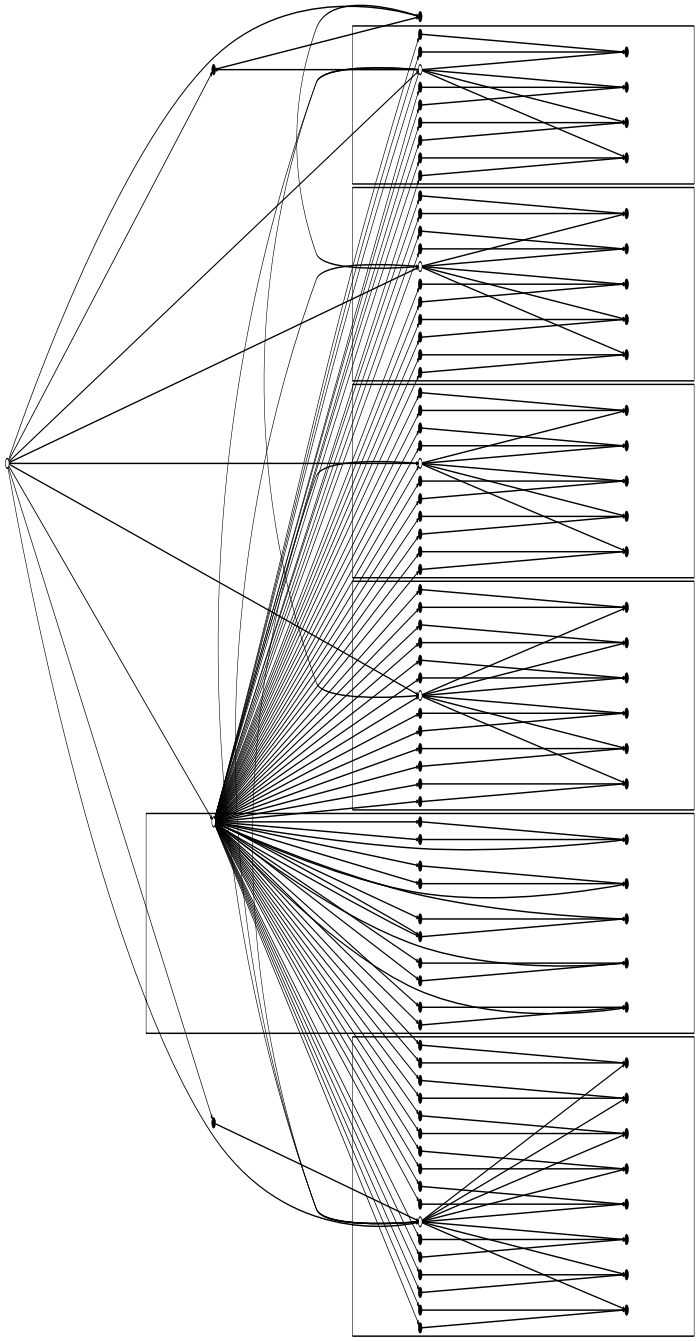


Figure 7.5: The complete structure of the credal network. Black nodes denote manifest variables, while latent variables are white. Boxes are used to highlight the different subnetworks modeling the observations of the latent variables as in Figure 7.4.

7.4 Quantitative Assessment of the Network

According to the discussion in Section 2.4, the specification of a credal network over the variables associated to the directed acyclic graph in Figure 7.5 requires the specification of a conditional credal set for each variable and each possible configuration of its parents.

For the core variables, these credal sets have been obtained by means of probability intervals explicitly provided by the Expert (Section 7.4.1), while, regarding observations, presence and reliability, a quantification procedure to automatically transform Expert's qualitative judgments in conditional credal sets specifications has been developed (Section 7.4.2).

7.4.1 Quantification of the Network Core

Because of the scarcity of historical cases, the quantification of the conditional credal sets for the core variables in Figure 7.2 is mainly based upon military and technical considerations. Together with the Expert we have isolated a number of principles, later translated into probability intervals and hence into conditional credal sets according to the procedure outlined in Section 2.3. As an example of the principles used to quantify this part of the network: *erroneous intruders are usually light aircrafts*, or *we do not expect a business jet or an airliner to be an erroneous intruder.*

In some situations, the Expert was also able to identify logical constraint among the variables. As an example, the fact that *balloons cannot maintain high levels of height* represents a constraint between the possible values of the variables AIRCRAFT TYPE and AIRCRAFT HEIGHT. These kinds of constraints have been embedded in the structure of the network by means of zero probability assessments.

7.4.2 Observations, Presence and Reliability

To complete the quantification of our credal network, we should discuss, for each sensor, the quantification of the variables associated to the observation, the reliability and the presence.

We begin by explaining how presence and reliability are specified. Consider the network in Figure 7.3. The Expert should quantify, for each of the four possible values of the variable AIRCRAFT HEIGHT, a credal set for the reliability and a credal set for the presence of the sensor. In practice, the Expert is simply required to suggest a value for the presence and a value for the reliability. To assess the value of the presence, he should take into consideration only the structure of

the identification architecture; while to assess the value for the reliability level, also the actual meteorological and geographical situation should be considered.

For each specified level of presence or reliability, the Expert should also decide whether or not he is uncertain about this value. His judgments are then translated into coherent probability intervals (see Section 2.3.3), from which we can compute the corresponding credal sets reflecting his beliefs. To this purpose, we have defined, together with the Expert, a set of fixed credal sets that are used to model the different combinations of values and uncertainty values. This procedure substantially simplifies the quantification of the network, while maintaining a large flexibility in the specification of presence and reliability.

Regarding the observations, a conditional credal set for each possible value of the corresponding latent variable and for each possible level of reliability and presence has been assessed. The idea is to avoid that the Expert would answer questions like, *what is the probability (interval) that the ground-based observers have medium reliability in observing the type of aircraft of an intruder that is flying at low height, if the meteorological condition is characterized by dense low clouds and we are in the plateau?* In fact, it can be extremely difficult and time-consuming to answer dozens of questions of this kind in a coherent and realistic way. It is much easier to answer questions like the following, *what is the reliability level that you expect from ground-based observers observing the type of aircraft of an intruder that is flying at low height, if the meteorological condition is characterized by dense low clouds and we are in the plateau?* The latter question is much simpler than the former, because one is required to specify something more qualitative than probabilities. This is exactly the type of question that we asked the Expert to quantify the necessary probabilities in our network. In the following we explain, in order, our quantification of presence and reliability of sensors and the observation mechanism.

Let X be a latent variable, and O the manifest variable corresponding to the observation of X as returned by a given sensor. For each possible joint value of RELIABILITY and PRESENCE, we should assess $\underline{P}(O = o|X = x)$ and $\overline{P}(O = o|X = x)$, for each $x \in \Omega_X$ and $o \in \Omega_O = \Omega_X \cup \{*\}$.

This quantification step can be simplified by a symmetric non-transitive relation of *similarity* among the elements of Ω_X. The *similarities* between the possible values of a latent variable according to a specific sensor can be naturally represented by an undirected graph as in the example of Figure 7.6. In general, given a latent variable X, we ask the Expert to determine, for each possible outcome $x \in \Omega_X$, the outcomes of X that are similar to x and those that are not similar to x.

Having defined, for each latent variable and each corresponding sensor, the similarities between its possible outcomes, we can then divide the possible obser-

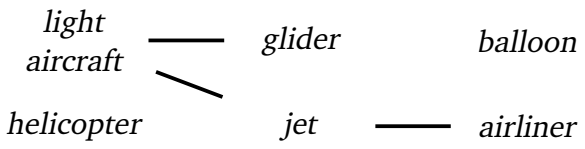


Figure 7.6: An undirected graph depicting similarity relations about the possible values of the variable TYPE OF AIRCRAFT according to the observation of a TV camera. Edges connect similar states. The sensor can mix up a light aircraft with a glider or a business jet, but not with a *balloon* or a *helicopter*.

vations in four categories: (i) observing the correct value of X; (ii) confounding the real value of X with a similar one; (iii) confounding the true value of X with a value that is not similar; (iv) the observation is *missing*. The idea is to quantify, instead of a probability interval for $P(O = o|X = x)$ for each $x \in \Omega_X$ and each $o \in \Omega_O$, only four probability intervals, corresponding to the four categories of observations described above. As an example, Table 7.1 reports an interval-valued quantification of the conditional probability table $P(O|X)$ for the ideal variable TYPE OF AIRCRAFT, for a combination of values of PRESENCE and RELIABILITY that models a good (although not perfect) quality of the observation.

	l. aircr.	*glider*	*balloon*	*helicopt.*	*jet*	*airliner*
l. aircraft	$[.9,.1]$	$[0,.1]$	0	0	$[0,.1]$	0
glider	$[0,.1]$	$[.9,1]$	0	0	0	0
balloon	1	0	$[.9,1]$	0	0	0
helicopter	0	0	0	$[.9,.1]$	0	0
jet	$[0,.1]$	0	0	0	$[.9,1]$	$[0,.1]$
airliner	0	0	0	0	$[0,.1]$	$[.9,.1]$
missing	$[0,.1]$	$[0,.1]$	$[0,.1]$	$[0,.1]$	$[0,.1]$	$[0,.1]$

Table 7.1: A model of a good quality observation of the AIRCRAFT TYPE, according to the similarity graph in Figure 7.6. A fixed probability interval $[0,.1]$ is assessed for the value *missing* and for the similar states.

Let us finally explain how the four probability intervals are quantified in our network for each combination of *reliability* and *presence* and for each sensor. The probability interval assigned to the case where the observation is missing depends uniquely on the *presence*. In particular, if the sensor is ABSENT, then the probability of having a MISSING observation is set equal to one and therefore the probability assigned to all the other cases are equal to zero. It follows that we have only seven combinations of *reliability* and *presence* to quantify. To this

end, we use constraints based on the concept of *interval dominance* to characterize the different combinations.[7] In order of accuracy of the observation, the combinations are the following:

(i) *high*, *present*: the correct observation dominates (clearly) the similar observations. The probability for not similar observations is zero and is therefore dominated by all the other categories.

(ii) *high*, *partially present*: the correct observation dominates the similar observations and dominates (clearly) the not similar observations. The similar observations dominates the not similar observations.

(iii) *medium*, *present*: the correct observation dominates the similar observations and dominates the not similar observations. The similar observations dominates the not similar observations.

(iv) *medium*, *partially present*: the correct observation does not dominate the similar observations but dominates the not similar observations.

(v) *low*, *present*: no dominance at all.

(vi) *low*, *partially present*: no dominance at all, but more overlapping among the intervals than in (5).

(vii) *absent* (no matter what the reliability is): the probability of a *missing* observation is equal to one, this value dominates all the other values.

7.5 Information Fusion by Imprecise Probabilities

In this section we develop an imprecise-probability approach to the general *information fusion* problem.

Let us first formulate the general problem. Given a latent variable X, and the manifest variables $O_1, \ldots, O_n$ corresponding to the observations of X returned by n sensors, we want to update our beliefs about X, given the values $o_1, \ldots, o_n$ returned by the sensors.

The most common approach to this problem is to assess a (precise) probabilistic model over these variables and compute the conditional mass function $P(X|o_1, \ldots, o_n)$. That may be suited to model situations of *consensus* among the different sensors. The precise models tend to assign higher probabilities to the

[7] Given a credal set $K(X)$ over a random variable X, and two possible values $x, x' \in \Omega_X$, we say that the x *dominates* x' if $\underline{P}(X = x') < \underline{P}(X = x)$ for each $P \in K(X)$. It is easy to show that that interval dominance, i.e., $\overline{P}(X = x') < \underline{P}(X = x)$, is a sufficient condition for dominance.

values of X returned by the majority of the sensors, which may be a suitable mathematical description of these scenarios.

The problem is more complex in case of *disagreement* among the different sensors. In these situations, precise models assign similar posterior probabilities to the different values of X. But a flat posterior probability mass function models *indifference*, while sensors disagreement seems to reflect instead a condition of *ignorance* about X.

Imprecise-probability models are more suited for these situations. Posterior ignorance about X can be represented by the impossibility of a precise specification of the conditional mass function $P(X|o_1,\ldots,o_n)$. The more disagreement we observe among the sensors, the wider we expect the posterior intervals to be, for the different values of X.

The case where the size of the posterior probability intervals results to be increased by conditioning is known in literature as *dilation* [SW93], and is relatively common with coherent imprecise probabilities.

The following simple example, despite its simplicity, is sufficient to outline how these particular features are obtained by our approach.

Example 1. *Consider a credal network over a latent variable X, and two manifest variables O_1 and O_2 denoting the observations of X returned by two identical sensors. Assume to be given the strong independencies coded by the graph in Figure 7.7. Let all the variables be Boolean. Assume $P(X)$ to be uniform and both $P(O_i = \mathrm{T}|X = \mathrm{T})$ and $P(O_i = \mathrm{F}|X = \mathrm{F})$ to take values in the interval $[1-\epsilon, 1]$, for each i=1,2, where $\epsilon > \frac{1}{2}$ models a (small) error in the observation mechanism. Since the network in Figure 7.7 can be regarded as a* naive credal classifier *[Zaf02], where the latent variable X plays the role of the class node and the observations correspond to the class attributes, we can exploit the algorithm presented in [Zaf02, Section 3.1] to compute the following posterior interval:*

$$P(X = \mathrm{T}|O_1 = \mathrm{T}, O_2 = \mathrm{T}) \in [\frac{(1-\epsilon)^2}{1-2\epsilon(1-\epsilon)}, 1].$$

It follows that, in case of consensus between the two sensors, the corresponding probability for the latent variable increases, given that the lower extreme is larger than $\frac{1}{2}$. In the case of disagreement, instead, we obtain that $P(X = \mathrm{T}|O_1 = \mathrm{F}, O_2 = \mathrm{T}) \in [0, 1]$, which means that our ignorance about X dilates, leading to a completely uninformative posterior interval.

Remarkably, assuming fixed levels of height, reliability and presence, Figure 7.4 reproduces the same structure of the prototypical example in Figure 7.7, with four sensors instead of two. The same holds for any sub-network modeling the relations between a latent variables and the relative manifest variables in our network.

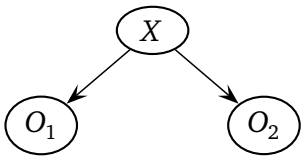


Figure 7.7: The credal network for Example 1.

7.6 Simulations

The discussion in Section 7.3 and Section 7.4 led us to the specification of a credal network, associated to the graph in Figure 7.5, over the whole set of random variables we consider, i.e., core variables, observations collected by the different sensors, reliability and presence levels.

At this point, we can evaluate the risk associated to an intrusion, by simply updating the probabilities for the four possible values of the risk factor, conditional on the values of the observations returned by the sensors and on the levels of reliability and presence observed by the Expert.

As a preliminary test of the model, we have considered a simulated scenario of a restricted flight area for the protection of a single object in the Swiss Alps, surveyed by an identification architecture that is characterized by the absence of interceptors and by a relatively good coverage of all the other sensors. We assumed as meteorological conditions discontinuous low clouds and daylight. The simulated scenario reproduces a situation where an agent provocateur is flying very low with a helicopter and without emitting any identification code. The decision maker is assumed to have uniform prior beliefs about the four classes of risk.

The size of the network suggests the opportunity of an approximate approach to this updating problem. In our approach, we have first augmented our credal network by a number of control nodes according to a *decision-theoretic* specification, according to the procedure developed in Chapter 4. Then, we have transformed each non-binary variable of the credal network into a set of binary variables, according to the GL2U algorithm reported in Chapter 5. In our case, the credal network has been updated in few seconds on a 2.8 GHz Pentium 4 machine, and convergence of L2U has been observed after seven iterations.

Figure 7.8.a depicts the posterior probability intervals for this simulated scenario. The upper probability for the outcome *renegade* is zero, and we can therefore exclude a terrorist attack. Similarly, the lower probability for the outcomes *agent provocateur* and *damaged intruder* are strictly greater than the upper probability for the state *erroneous*, and we can reject also this latter value because

of interval dominance. Both these results are reasonable estimates for this simulated scenario.

Remarkably, the indecision between *agent provocateur* and *damaged intruder* disappears as we assume higher levels of reliability and presence for the sensors devoted to the observation of the *height*. The results, reported in Figure 7.b, state that the intruder is an *agent provocateur*, as we have assumed in the design of this simulation.

As a final comment on these simulations, we have experienced a substantial agreement between the estimates provided by the credal networks and those returned by the military experts for the same scenarios. Nevertheless, for sake of fairness, it should be pointed out that, at the present moment, our model is the result of a number of interactions with the military experts and has been designed with the aim of meet their judgements in the considered scenarios. A deeper validation of the quality of the estimated provided by the credal network should be therefore considered as a necessary future work.

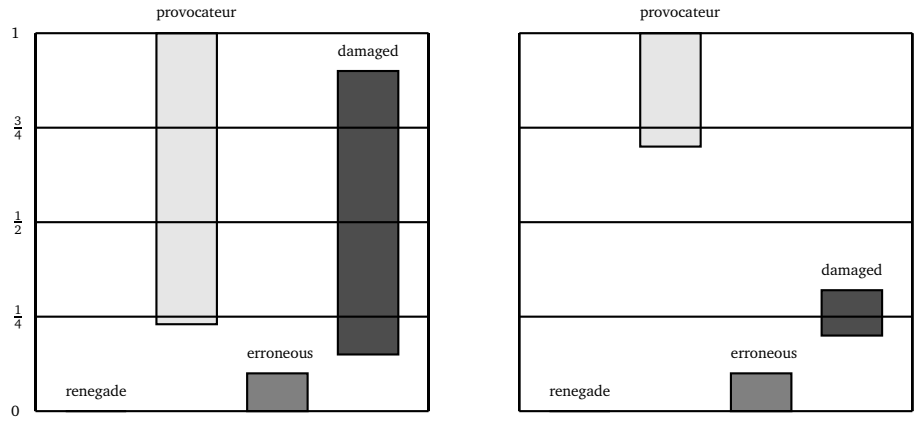


Figure 7.8: Posterior probability intervals for the risk factor, corresponding to a simulated scenario reproducing a helicopter entering the restricted flight area for demonstrative reasons. The histogram bounds denote lower and upper probabilities. The quality of the observation of the AIRCRAFT HEIGHT is assumed to be higher in (b) than in (a).

7.7 Summary and Outlooks

A model for determining the risk of intrusion of a civil aircraft into restricted flight areas has been presented. The model embeds in a single coherent math-

ematical framework human expertise expressed by imprecise-probability assessments, and a structure reproducing complex observation mechanisms and corresponding information fusion schemes.

The risk evaluation corresponds to the updating of the probabilities for the risk factor conditional on the observations of the sensors and the estimated levels of presence and reliability. Preliminary tests considered for a simulated scenario are consistent with the judgments of an domain expert for the same situation.

As a comment, it seems possible to offer a practical support to the military experts in their evaluations. They can use the network to decide the risk level corresponding to a real scenario, but it is also possible to simulate situations and verify the effectiveness of the different sensors in order to design an optimal identification architecture.

Finally, we regard our approach to the fusion of the information collected by the different sensors as a sound and flexible approach to this kind of problems, able to work also in situations of contrasting observations between the sensors.

With respect to future work, we intend to test the model for other historical cases and simulated scenarios. The approximate updating procedure considered in the present work, as well as other algorithmic approaches will be considered, in order to determine the most suited for this specific problem. In any case, it seems already possible to offer a practical support to the military experts in their evaluations. They can use the network to decide the risk level corresponding to a real scenario, but it is also possible to simulate situations and verify the effec- tiveness of the different sensors in order to design an optimal identification architecture.

Chapter 8

Credal networks for Hazard Assessment of Debris Flows

Debris flows (Section 8.1) are among the most dangerous and destructive natural hazards that affect human life, buildings, and infrastructures. Starting from the '70s, significant scientific and engineering advances in the understanding of the processes have been achieved [CW87; IRL97]. Yet, human expertise is still fundamental for hazard identification as many aspects of the whole process are still poorly understood.

In Section 8.2 we try to fill the modeling gap by using a separately specified credal network. According to the discussion in Section 2.4.2, we capture the causal relationships between the triggering factors of debris flows by a directed graph, and we represent quantitative influences by probability intervals, determined from historical data, expert knowledge, and theoretical models. The model presented aims at supporting experts in the prediction of dangerous events of debris flow. It is worth emphasizing that the credal network model joins human expertise and quantitative knowledge; this seems to be a necessary step for drawing credible conclusions. We are not aware of other approaches with this characteristic.

In Section 8.2.3 we present preliminary experiments testing the model on historical cases of debris flows happened in the Ticino canton. The case studies highlight the good capabilities of the model: for all the areas the model produces significant probabilities of hazard. We make a critical discussion of the results, showing how the results are largely acceptable by a domain expert. Finally, in Section 8.2.4, with the support of a detailed GIS analysis, we test this procedure for a whole, debris flow prone, watershed. The results indicate that the model detects the areas of the basin more prone to debris flow initiation and produces

different hazard patters according to different rainfall events.[1]

8.1 Debris Flows

Debris flows are gravity-induced mass movement intermediate between landslides and water floods. They are composed of a mixture of water and sediment with a characteristic mechanical behavior varying with water and soil content. Three types of debris flow initiation are relevant: erosion of a channel bed due to intense rainfall, landslide, or destruction of a previously formed natural dams. According to [CF84], prerequisite conditions for most debris flows include an abundant source of unconsolidated fine-grained rock and soil debris, steep slopes, a large but intermittent source of moisture (rainfall or snowmelt), and sparse vegetation. As mentioned in [GWM04], several investigation have focused on debris flows initiation and frequency. Among them, [Gla05] focused on existing links between debris-flow hazard and geomorphology. Several hypotheses have been formulated to explain mobilization of debris flows and this aspect still represents a research field. The identification procedure presented here is based on the theoretical model proposed by [Tak91], although a different explanation of the triggering of debris flow by channel-bed failure has been recently described by [AG05].

The mechanism to disperse the materials in flow depends on the properties of the materials (like that grain average size, which is also called *granulometry* and the internal friction angle), channel slope, flow rate and water depth, particle concentration, etc., and, consequently, the behavior of flow is also various. Unfortunately, not all the triggering factors considered by this model can be directly observed, and their causal relations with other observable quantities can be shaped only by probabilistic relations, by means of the formalism introduced in the following section.

8.2 The Credal Network

8.2.1 Causal Structure

The network in Figure 8.1 expresses the causal relationships between the topographic and geological characteristics, and hydrological preconditions. The leaf node MOVABLE DEBRIS THICKNESS is the depth of debris likely to be transported downstream during a flood event. Such node represents an integral indicator of the hazard level. Here we describe the considerations that led to such graph.

[1] The work presented in this chapter has been done in cooperation with Andrea Salvetti.

Debris flows require a minimum thickness of *colluvium* (loose, incoherent deposits at the foot of steep slope) for initiation, produced from a variety of bedrock. This is embedded in the graph with the connection to the node AVAILABLE DEBRIS THICKNESS and expresses the propensity of the rock to produce sediment.

On the other side, the node GEOLOGY represents the characteristics of the bedrock in a qualitative way. Bedrock properties influence the rate of infiltration and deep percolation, so affecting the generation of surface runoff and the concentration in the drainage network. This is accounted for by the connection of the geology to the HYDROLOGIC SOIL TYPE, which influences the MAXIMUM SOIL WATER CAPACITY.

The SOIL PERMEABILITY, i.e. the rate at which fluid can flow through the pores of the soil, has to be further considered. If permeability is low, the rainfall tends to accumulate on the surface or flow along the surface if it is not horizontal. The causal relation among geology and permeability determining the different hydrologic soil types was adopted according to [Kun02]. The basic assumption is that soils with high permeability and extreme thickness show a high infiltration capacity, whereas shallow soils with extremely low permeability have a low infiltration capacity.

The LAND USE cover of the watershed is another significant cause of debris movement. It characterizes the uppermost layer of the soil system and has a definite bearing on infiltration.

The *curve number method* [Ser93] has been adopted in order to define the infiltration amount of precipitation, i.e. the MAXIMUM SOIL WATER CAPACITY. This method distinguishes hydrologic soil types which are supposed to show a particular hydrologic behavior. For each land use type there is a corresponding curve number for each hydrologic soil type.

The amount of rainfall which cannot infiltrate is considered to accumulate into the drainage network surface runoff, increasing the WATER DEPTH and eventually triggering a debris flow in the river bed.

These processes are described by the deterministic part of the graph, related to runoff generation and Takahashi's theory, which takes into account topographic and morphologic parameters, such as LOCAL SLOPE of the source area, watershed morphology (described by the BASIN RESPONSE FUNCTION), UPSTREAM CONTRIBUTING AREA, RAINFALL INTENSITY, and CHANNEL WIDTH. Regarding the width of the channel, it should be pointed out that the complexity of the channel geometry is usually low and almost similar in debris flow prone watersheds. For this reasons it was decided to adopt only three categories of channel width.

The channel width is obviously decisive to determine the water depth, given the runoff generated within the watershed according to the standard hydraulic

assumptions. Field experience in the study region indicates that debris flows often start in very steep and narrow creeks, with reduced accumulation area upstream.

The climate of the regions in which debris flows are observed is as varied as geology and this was accounted for by defining several climatological regions, with different parameters of the depth-duration-frequency curve. In addition to the RAINFALL DURATION and EFFECTIVE RAINFALL INTENSITY of a storm that ultimately produces a debris flow, the ANTECEDENT SOIL MOISTURE conditions are recognized as an important characteristic. The significant period of antecedent rainfall varies from days to months, depending on local soil characteristics. According to the curve number theory, the transformation law to the EFFECTIVE SOIL WATER CAPACITY depends only on the five-days antecedent rainfall amount corresponding to different moisture conditions.

We used the *linear theory of the hydrologic response* to calculate the MAXIMUM PEAK RUNOFF values produced by constant-intensity hyetographs. We used the *multiscaling framework for intensity duration frequency curve* [BR96] coupled with the *instantaneous unit hydrograph* theory, proposed by [RCP+04]. Accordingly, the time to peak is greater than the rainfall duration and the CRITICAL RAINFALL DURATION is independent of rainfall return period. The instantaneous unit hydrograph was obtained through the *geomorphological theory* [RIV79] and the *Nash cascade model of catchment's response*, where the required parameters were estimated from HORTON'S RATIOS according to [Ros84].

By using the classical river hydraulics theory, the water depth in a channel with uniform flow and given discharge, water slope and roughness coefficient can be determined with the *Manning-Strickler formula* [Mai93]. The granulometry is required to apply Takahashi's theory. The friction angle was derived from the granulometry with an empirical one-to-one relationship. Takahashi's theory can finally be applied to determine the THEORETICAL DEBRIS THICKNESS that could be destabilized by intense rainfall events. The resulting value is compared with the actual AVAILABLE DEBRIS THICKNESS in the river bed. The minimum of these two values is the MOVABLE DEBRIS THICKNESS.

8.2.2 Quantification

Quantifying uncertainty means to specify the conditional credal sets for all the nodes, given all the possible instances of their parents. The specification is imprecise, in the sense that each conditional probability can lie in an interval. Intervals were inferred for the nodes GEOLOGY, PERMEABILITY, LAND USE, LOCAL SLOPE, HYDROLOGIC SOIL TYPE, and MAXIMUM SOIL WATER CAPACITY, from the GEOSTAT database [KKW01], by the imprecise Dirichlet model (see Section 2.3.4). The

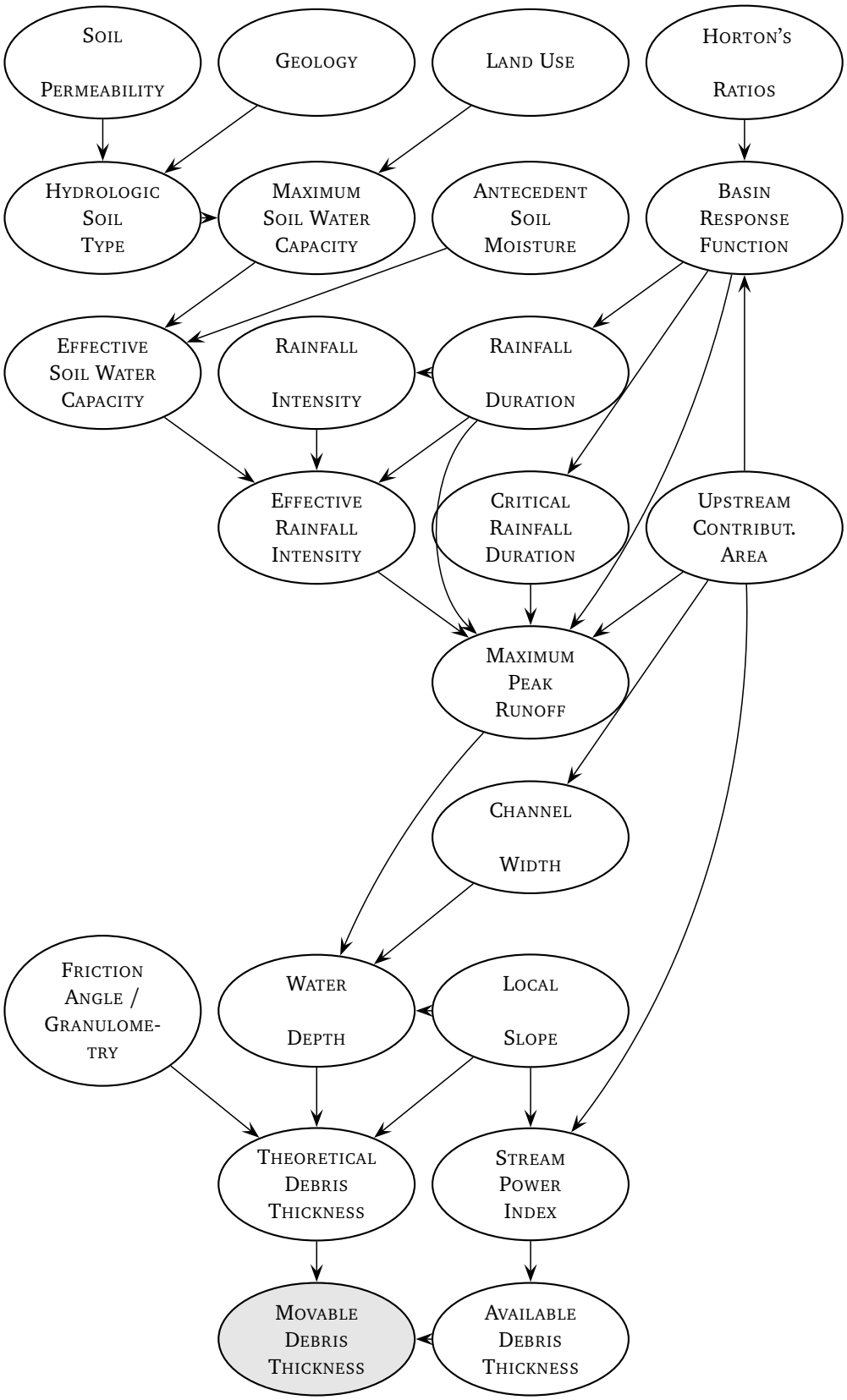


Figure 8.1: The credal network for hazard identification.

expert provided intervals for nodes CHANNEL WIDTH, GRANULOMETRY, HORTON'S RATIOS, and AVAILABLE DEBRIS THICKNESS. Functional relations between a node and its parents were available for the remaining nodes; in this case the intervals degenerate to a single 0-1 valued mass function. We detail the functional part in the rest of the section.

As mentioned in Section 8.1, the ANTECEDENT SOIL MOISTURE conditions were accounted for by using the curve number method. The parametrization (b_1, b_2) of the BASIN RESPONSE FUNCTION corresponding to the instantaneous unit hydrograph was obtained by using the number of theoretical linear reservoirs by which the basin is represented, $b_1 = 3.29 \cdot r_1^{0.78} \cdot r_2^{0.07}$; and by the time constant of each reservoir, $b_2 = .7 \cdot 0.251 \cdot (r_1 \cdot r_2)^{-.48} \cdot a^{0.38}$. Here b_1 depends on *Horton's ratios* r_1 and r_2, and b_2 is also function of the UPSTREAM CONTRIBUTING AREA a. For this we assumed the empirical expression reported by [DR03].

Given b_1 and b_2, following [RCP+04], we calculate the two characteristic durations, the RAINFALL DURATION t and CRITICAL RAINFALL DURATION t', by solving the following system of two equations:

$$\begin{cases} \alpha = [\frac{t}{b_2} \cdot (\frac{t'}{b_2})^{b_1-1} e^{-t'/b_2}] / [\gamma(b_1, \frac{t'}{b_2}) - \gamma(b_1, \frac{t'-t}{b_2})] \\ \frac{t}{t'} = 1 - e^{-\frac{t}{b_2} \cdot \frac{1}{b_1-1}} \end{cases}, \tag{8.1}$$

where γ is the *incomplete lower gamma function* and α is a parameter, corresponding to the exponent of the *multiscaling intensity duration frequency curve*.
We assume that these are in the form $i' = f(\tau_r) \cdot t^{-\alpha}$, where f is function of the return period τ_r of the event and i' is the RAINFALL INTENSITY. To evaluate the EFFECTIVE RAINFALL INTENSITY i, we have to impose the following transformation, taking account of the (effective) curve number, the corresponding dispersion term, and of the rainfall duration:

$$i = \frac{(i' \cdot t - \lambda(c)/10)^2}{i' \cdot t - \lambda(c)/10} + \lambda(c) \cdot 1/t, \tag{8.2}$$

where $\lambda(c) = 254 \cdot (100/c - 1)$ is the water depth absorbed by the soil of given curve number. The maximum runoff along the drainage network was calculated according to the well-established theory of the *Instantaneous Unit Hydrograph* (IUH), expressed as the convolution integral of the effective rainfall input. The hydrograph shape strongly depends on the geomorphological features of the river basin, therefore, the *Geomorphologic Instantaneous Unit Hydrograph* (GIUH) presents obvious advantages in ungauged watersheds [RIV79], since the GIUH only depends on the morphological characteristics of the watershed and the drainage network. According to this theory, the MAXIMUM PEAK RUNOFF q is

obtained using the following:

$$q = \begin{cases} i \cdot a \cdot [H(t') - H(t' - t)] & 0 \leq t \leq \tau_c \\ \\ i \cdot a & t > \tau_c, \end{cases} \tag{8.3}$$

where $H(t)$ represents the integral of the GIUH from the beginning of the storm, t^* is the critical duration at the considered point, and it is a function of the RAINFALL DURATION t, while τ_c is the concentration time. Effective rainfall intensity is determined using the well established *SCS Curve Number* infiltration method and the rainfall intensity modeled by multiscaling power law relationship. The CRITICAL RAINFALL DURATION t' associated with the extreme peak runoff is independent of the return period and of the rainfall intensity; the corresponding rainfall volume is calculated for the rainfall duration t. The corresponding WATER DEPTH is

$$w = \frac{q}{25} \cdot l^{\frac{5}{3}} \cdot \sqrt{\tan n}, \tag{8.4}$$

where n is the LOCAL SLOPE and l the CHANNEL WIDTH. According to [Tak91], we evaluate the THEORETICAL DEBRIS THICKNESS as

$$d' = w \cdot \left[k \cdot (\frac{\tan m'}{\tan n} - 1) - 1 \right]^{-1}. \tag{8.5}$$

The relation is linear, with a coefficient taking into account the local slope n and the FRICTION ANGLE m' (which can be obtained from the GRANULOMETRY m). $k = C_g(\delta_g - 1)$, with $\delta_g = 2.65$ the relative density of the grains, and $C_g \simeq 0.7$ the volumetric concentration of the sediments. The effect of a water depth on the movable debris quantity is based on the equilibrium of forces acting on a debris cluster under different conditions. According to [Tak91], the local slope for which debris-flow formation can take place obeys the following constraint:

$$\frac{C_g \delta_g}{\frac{4}{3} + C_g \delta_g} \tan m' \leq \tan n \leq \frac{C_g \delta_g}{1 + C_g \delta_g} \tan m', \tag{8.6}$$

For the points of the basin whose values of m and n do not satisfy the constraint in Equation (8.6), either the cluster is not completely saturated and, if unstable at high slope angles, produces a landslide or the process that takes place is the ordinary solid transport ([DBA06]), and therefore we drop the relative point from the potential source areas of this hazard without any further analysis.

The theoretical value for the movable quantity d' does not take into account how much material is physically available. As the actual MOVABLE DEBRIS THICKNESS d cannot exceed the AVAILABLE DEBRIS THICKNESS x, the final relation is

given by

$$d = \min\{x, d'\}. \tag{8.7}$$

Many authors have dealt with the capability of *local slope* and *upstream contributing area* to account for topographic control on erosion and deposition potential in complex terrain and with the use of slope and contributing area for channel network extraction, based on critical area and slope-area threshold (e.g., [PA96]). In this study such a method was used to extract the channelized portion of the *Digital Elevation Model* (DEM), where debris flow initiation can appear, according to the following equation:

$$SPI = \sqrt{A} \cdot \theta, \tag{8.8}$$

where SPI denotes the STREAM POWER INDEX. The threshold value of SPI has been identified by trials, comparing the extracted network with the drainage network on the map, where also many ephemeral channel in the upper part of the basin were included in the network. That index can be used as an indicator of the local transport capacity of a single reach along the network and, therefore, to identify channel reach were debris material preferentially accumulates. Clearly, the availability of an abundant debris thickness in the drainage network is a fundamental precondition for debris initiation. Based on some previous work of [DFM03], we developed a conceptual framework for a qualitative evaluation of the debris availability in the river network. We assume that the debris availability is a function of the convenience capacity of the river network associated to the SPI. Cells with SPI value exceeding the threshold for channel initiation correspond very often to areas where bedrocks emerge and local slope is quite high, and therefore the sediment deposition is zero or very low. On the contrary, in cells where SPI is much less than the selected threshold level, high deposition instead of erosion is expected and we therefore assume a high availability of debris material. These principles supported by expert knowledge have been used for an interval-valued probabilistic quantification of the node AVAILABLE DEBRIS THICKNESS.

According to the model of the initiation mechanism considered in this study, the soil failure is induced by surface runoff and, consequently, the maximum discharge and the corresponding water depth must be estimated. [RIR97] investigated how the variation of the characteristics of stream channel is expressed as a function of the discharge by a power law at a given cross section and also along the channel network. The parameters were estimated by using a few collected cross-section data, randomly distributed along the drainage network.

8.2.3 Local Identifications

We validate the model in preliminary way by an empirical study involving six areas of the Ticino canton. The network was initially fed with the information about the areas reported in Table 8.1, the estimated rainfall intensity on them for a return period of 10 years, and the geomorphological characteristics of the watershed. The estimated rainfall intensity is the expected frequency level of precipitations in a certain region during a future period. Using the estimated rainfall intensity allowed us to re-create the state of information existing 10 years ago about precipitations in the areas under consideration. This is a way to check whether the network would have been a valuable tool to prevent the debris flows that actually happened in the six areas.

Node	Cases					
	1	2	3	4	5	6
G	Gneiss	Porphyry	Limestone	Gneiss	Gneiss	Gneiss
A	0.26	0.32	0.06	0.11	0.38	2.81
M	10–100	$\leq$10	$\leq$10	100–150	$\leq$10	150–250
U	Forest	Forest	Forest	Vegetation	Forest	Bare soil
N	20.8	19.3	19.3	21.8	16.7	16.7
L	4	6	4	8	4	8
R_1	0.9	0.6	0.7	0.9	0.9	0.8
R_2	1.5	3.5	3.5	3.5	2.3	2.1

Table 8.1: Details about the six case studies. Note that the PERMEABILITY is unavailable. This is a common case because of the technical difficulties in its evaluation.

Thickness	Cases					
	1	2	3	4	5	6
low	0.011	[0.084,0.087]	0.083	0.196	0.087	0.005
medium	0.048	[0.263,0.273]	0.275	0.388	0.139	0.013
high	0.941	[0.639,0.652]	0.642	0.416	0.774	0.982

Table 8.2: Posterior probabilities for MOVABLE DEBRIS THICKNESS. The probabilities are displayed by intervals in case 2.

The results of the analysis are in Table 8.2. We use the probabilities of defined debris thickness to be transported downstream as an integral indicator of the hazard level.

In cases 1 and 6 the evidences are the most extreme out of the six cases and indicate a high debris flow hazard level, corresponding to an unstable debris thickness greater than 50 cm. In case 6 the relatively high upstream area (2.81 km^2), large channel depth, and the land cover (bare soil, low infiltration capacity) explain the results. In case 1 the slope of the source area (20.8°) plays probably the key role. In cases 2 and 3 the model presents a non-negligible probability of medium movable debris thickness. Intermediate results were obtained for case 5 due to the gentler bed slope (16.7°) as compared with the other cases. In case 4 the hazard probability is more uniformly distributed, and can plausibly be explained with the very small watershed area and the regional climate, which is characterized by low small rainfall intensity as compared with other regions.

We simulated also the historical events, by instantiating (as opposed to using the estimated rainfall intensity) the actual measured rainfall depth, its duration and the antecedent soil moisture conditions. Also in this setup the network produced high probabilities of significant movable thickness.

As more general comment, it is interesting to observe that in almost all cases the posterior probabilities are nearly precise. This depends on the strength of the evidence given as input to the network about the cases, and by the fact that the flow process can partially be (and actually is) modeled functionally.
Now we want to model the evidence in even more realistic way with respect to the grain size of debris material. Indeed, granulometry is typically known only partially, and this limits the real application of physical theories, also considered that granulometry is very important to determine the hazard.

We model the fact than the observer may not be able to distinguish different granulometries by the *conservative inference rule* described in Section 4.4. To this end, we add a new node to the net, say O_M, that becomes parent of M. O_M represents the observation of M. There are five possible granulometries, m_1 to m_5. We define the possibility space for O_M as the power set of $\mathscr{M} = \{m_1, \ldots, m_5\}$, with elements $o_{\mathscr{M}'}$, $\mathscr{M}' \subseteq \mathscr{M}$. The observation of granulometry is set to $o_{\mathscr{M}'}$ when the elements of $\mathscr{M}'$ cannot be distinguished. $P(m|o_{\mathscr{M}'})$ is defined as follows: it is set to zero for all states $m \in \mathscr{M}$ so that $m \notin \mathscr{M}'$; and for all the others it is *vacuous,* i.e. the interval $[0,1]$ (the intervals defined this way must then be made reachable). This expresses the fact that we know that $m \in \mathscr{M}'$, and nothing else.
Let us focus on case 6 for which the observation of grain size is actually uncertain. From the historical event report, we can exclude that node M was in state

m_1 or m_2. We cannot exclude that m_4 was the actual state (m_4 is the evidence used in the preceding experiments), but this cannot definitely be established. We take the conservative position of letting the states m_3, m_4 and m_5 be all plausible evidences by setting $O_M = o_{\{m_3,m_4,m_5\}}$. The interval probabilities become $[0.002, 0.008]$, $[0.010, 0.043]$, and $[0.949, 0.988]$, for debris low thicknesses, medium, and high, respectively. We conclude that the probability of the latter event is very high, in robust way with respect to the partial observation of grain size.

8.2.4 Spatially-Distributed Identifications

The case study we present in this section refers to the Acquarossa Creek in the Blenio Valley, an area located in the North-Eastern part of the Ticino Canton, Southern Switzerland. This area was selected because of the potential hazard caused by debris flows to communication lines and villages. That creek is a small tributary of the Brenno river, characterized by a high altitude range (from 530m up to 2580m a.s.l.) of the Simano Peak. Debris torrents are usually triggered by intense rainfall, following a period of abundant precipitation. Eight historical debris flow events were recorded in that area during the last 150 years. Most of them caused high damages to infrastructures on the alluvial fan, transporting several thousand cubic meters of material. For instance, during the last event in August 2003, a volume of about $15'000\text{m}^3$ were estimated on the alluvial fan, and a similar pattern was observed in 1983 and 1987. That represents a relatively high frequency of debris flow events. Accordingly, the triggering factors appear to be already effective in many parts of the basin with storm events of low and medium return period.

In order to gather evidential information about the geomorphological characteristics of the basin, a highly precise DEM based on airborne laser scanning produced by the Swiss Federal Office of Topography has been employed. That offers a spatial resolution of 4 meters, which is comparable with the typical channel width; that defines a drainage network of 6310 cells. Most of the morphological data used for our identification analysis (slope, flow-direction and flow-accumulation) were derived from this dataset, and the SPI was calculated as in Equation (8.8).

Finally, regarding the observation of the granulometry, a field survey was conducted. The river bed and lateral debris levees were analyzed in order to determine the grain-size distribution of the debris material. A significant difference was observed for the grain-size distributions obtained from several samples. We have therefore decided to split the basin into two sub-regions of "uniform" granulometry, and describe the outcome of the sampling by a *soft evidence* modeled

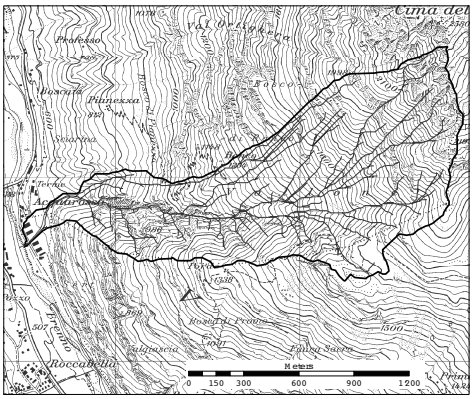


Figure 8.2: Acquarossa Creek Basin.

as a new unconditional credal sets for the corresponding node in the credal network.

In order to avoid unnecessary computations, for each point of the basin, we have preliminarily checked whether or not the observed slope and the values of the friction angle compatible with the soft observation of the granulometry were compatible with the constraint in Equation (8.6). This deterministic pre-analysis detects 170 pixels where only ordinary sediment transport is possible and 135 pixels that are already unstable without complete soil saturation. For the remaining 6005 pixels, we have computed the posterior lower and upper probabilities for the movable debris thickness corresponding to observed geomorphological factors and rainfall intensity for a return period of 10, 30 and 100 years. These computations have been exactly performed by exhaustive approaches based on the iteration of standard algorithms for Bayesian networks as our credal network is equivalent to about 500 Bayesian networks. The network is thus expected to predict the probability of a debris flow event with the defined frequency level at each point of the drainage network. In this way, we aim at verifying whether the network would have been a valuable tool to predict considerable events of debris flows, which actually happened in the areas under consideration, and, more important, to identify the points where the debris flow is most likely to occur in the future. Figure 8.3 reports the results of the inference process for respectively 10 and 100 years return period rainfall event.

We observe that the debris flow is more likely to initiate on the main channel, even in the lower part of the basin. This fits with the historical observations

for this basin and also for other watersheds in the same region. Regarding the role of the return period in our tests, we observe an increase of the number of dangerous points, that spread upstream along the drainage network: for higher return periods even a small upstream area is sufficient to produce a peak runoff that can trigger a debris flow. All these remarks, which refer to results obtained by an almost automatic procedure, are considered acceptable by an expert domain.

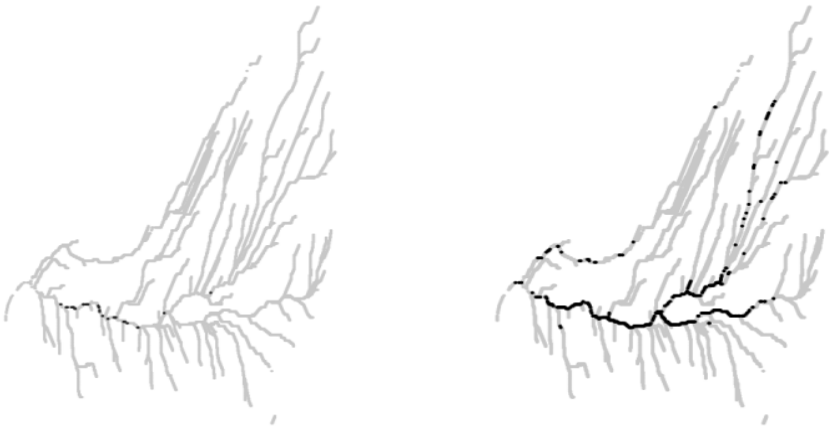


Figure 8.3: Spatially distributed identifications for the basin in Figure 8.2 and rainfall return periods of 10 (left) and 100 (right) years. The points for which the credal network predicts the lower class of risk are depicted in gray, while black refers to points where higher levels of risk cannot be excluded.

8.3 Human versus Artificial Expert

In order to evaluate the quality of the numerical tests reported in Section 8.2.3 and Section 8.2.4, we have asked a comment on these result to Andrea Salvetti, the expert of geomorphological natural hazards which has collaborated with us during the development of the model presented in this chapter. As a first general comment, all the estimates provided by our credal networks are fitting the previsions of our expert. It seems therefore possible to regard our model as a real *artificial expert system* able to replace, or at least support, "human" experts in the risk analysis of debris flows.

Regarding the historical cases in Section 8.2.3, it should be also pointed out that the posterior probability intervals returned by our credal network are nearly precise. This seems to be related to the fact that our model embed a number of deterministic relations, which reduce the imprecision of the inferences when a sufficient number of factors is observed. Despite the substantial agreement with the expert estimates, the human conclusions for the same cases are much more qualitative. As a possible comment on this issue, let us point out the relative toughness of human reasoning in managing on the same time probabilistic and deterministic knowledge in single process, while, on the other side, a probabilistic model can naturally embed deterministic relations by simply regarding them as degenerate mass functions specifications.

Similar considerations can be done for the area-distributed simulations. Considering that the this kind of analysis requires the independent computation of the level of risk in thousands of cells of the basin, it should be pointed out that a human expert would be able to perform a similar analysis in a considerably long time, while the credal network can return a risk map as in Figure X in few seconds. Accordingly, the artificial expert system should be regarded not only as a replacement (or support) for the human expert, but also as a sort of "super-expert" for which we can obtain an arbitrary number of replica, that can perform the same analysis of a human expert in considarably shorter time and, in principle, in a parallel way.

8.4 Summary and Outlooks

We have presented a model for determining the hazard of debris flows based on credal networks. The model unifies human expertise and quantitative knowledge in a coherent framework. This overcomes a major limitation of preceding approaches, and is a basis to obtain credible predictions, as shown by the experiments. Credible predictions are also favored by the soft-modeling made available by imprecise probability through credal sets.

The model was developed for the Ticino canton in Switzerland, but extension to other areas is possible by re-estimating the probabilistic information inferred from data, which has local nature. The identification procedure can be extensively applied to whole basins, and unnecessary computations are avoided for areas where the geomorphological conditions are not compatible with debris flow initiation. As a spatially distributed case study, we tested our model for a debris flow prone watershed in Southern Switzerland. The model detects the areas inside the basin more prone to debris flow initiation and also shows that different rainfall return periods produce different hazard patterns. That makes it possible to determine the return period of the critical rainfall that triggers de-

bris flow as a result of channel-bed failure in a specific point along the drainage network.

Chapter 9

Conclusions and Future Research

9.1 Main Results

We have in this thesis presented a number of graphical transformations providing equivalent representations for different probabilistic graphical models. The notion of *equivalent representation* is probably the actual focus of the entire thesis and has led to two different kinds of results. First, our equivalent representations offer alternative and sometimes more general and expressive languages of specification, leading to unified views of models previously considered different and irreconcilable. Second, and even more important, the connection established by these relations can be used to solve (and evaluate the complexity of) inference problems, which might not be so easy to do in their original formulations.

Therefore, in order to summarize the findings of this thesis, let us first report the equivalence relations we have established in the area of probablistic graphical models.

- The notion of *decision-theoretic* specification of a credal network (Section 4.1), which defines a new class of probablistic graphical models including both non-separately and separately specified credal networks, allowing for a unified representation of these two classes of models.

- An equivalence relation between credal networks specified in the decision-theoretic framework and separately specified credal networks defined over a wider domain (Section 4.2).

- The notion of *exact binarization* of a credal network, providing an equivalent representation of a credal network of any kind as a separately specified credal network defined only over binary variables (Section 5.2.1).

- The equivalence between Bayesian and credal networks with respect to a specific updating problem with incomplete observations (Section 3.1.2).

- The equivalnce between a Bayesian network and a *valuation algebra*, which is a more abstract class of models, with respect to a specfic classification problem, still involving missing observations (Section 6.5).

All these results can be regarded as important advances into the field of graphical models, offering a deeper and more general view of existing classes of models and their relations with other and even new classes. Further, these results have been also employed for the development of the following inference algorithms.

- A procedure that extends any algorithm designed for separately specified credal networks to credal networks of any kind (e.g., Section 4.3).

- The GL2U algorithm, which represents a state-of-the-art algorithm for general credal networks purely based on message propagation (Section 5.2.2).

We have also presented two applications of credal networks to real problems (Chapters 7 and 8). These are among the very first examples of real application of these mathematical models. Moreover, besides their intrinsic importance in solving the problems for which they have been designed, we regard these two networks as prototypical examples of applications based on credal networks, able to offer useful guidelines to other researchers aiming to develop similar applications.

9.2 Future Research Directions

Finally, let us point out some future research directions that are relevant for the work presented in this thesis.

Let us start from the theoretical issues. We point the reader to Sections 4.5, 5.3, and 6.7, where the possible developments of the specified results provided in the relative chapters are detailed. Here, let us consider this point from a more general perspective. The main directions opened by the findings proposed in this thesis are the following three.

- By means of the notion of decition-theoretic specification of a credal network we have developed a first connection between credal networks and decision graphs. A further development in this direction could be achieved with important results in terms of cross-fertilization for both these fields.

- The GL2U algorithm we have developed has been already presented as a state-of-the-art algorithm for approximate updating of large credal networks. Nevertheless, it seems possible to improve the performance of the algorithm, and also obtain a theoretical characterization of the accuracy and the convergence of the algorithm.

- We have modeled, by means of specific graphical transformation, the observation mechanism characterizing a specific updating problem, where the outcome of the observation is incomplete. This idea might be developed in order to describe other examples of uncertain observations. That would represent an important generalization to imprecise probabilities of the standard approaches to soft evidence modelling (e.g., Jeffrey's updating and virtual evidence method).

Concerning implementation issues, we intend to develop new software tools allowing an increasing number of users to work with credal networks. Most of the ideas outlined in this thesis could lead to the implementation of software tools of this kind.

First, the notion of decision-theoretic specification offers a new and general language for credal networks that could be implemented as an XML standard for the explicit specification of a credal network. Furthermore, the implementation of GL2U could represent the first step towards the development of a general framework for making inferences on credal networks. The development of a graphical interface simplifying the interaction with a credal network would be another important achievement, either for the applications presented in Chapter 7 and 8, and for any other application based on credal networks.

Bibliography

[AF96] D. Avis and K. Fukuda. Reverse search for enumeration. *Discrete Applied Mathematics*, 65:21–46, 1996.

[AG05] A. Armanini and C. Gregoretti. Incipient sediment motion at high slopes in uniform flow condition. *Water Resources Research*, W12431, 2005.

[BKRK97] J. Binder, D. Koller, S. Russell, and K. Kanazawa. Adaptive probabilistic networks with hidden variables. *Machine Learning*, 29(2-3):213–244, 1997.

[BMvH02] D. Boorsbom, G. J. Mellenbergh, and J. van Heerden. The theoretical status of latent variables. *Psychological Review*, 110(2):203–219, 2002.

[BR96] P. Burlando and R. Rosso. Scaling and multiscaling depth-duration-freqeuncy curves of storm precipitation. *Journal of Hydrology*, 177(1-2):45–64, 1996.

[BSCC89] I. Beinlich, H. J. Suermondt, R. M. Chavez, and G. F. Cooper. The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. In *II European Conference on Artificial Intelligence in Medicine*, pages 247–256, Berlin, 1989. Springer.

[CCM94] 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.

[CdCIFdR04] F. G. Cozman, C. P. de Campos, J. S. Ide, and J. C. Ferreira da Rocha. Propositional and relational Bayesian networks associated with imprecise and qualitative probabilistic assessments. In

M. Chickering and J. Halpern, editors, *Proceedings of the 20th Annual Conference on Uncertainty in Artificial Intelligence (UAI 2004)*, pages 104–111, Arlington, Virginia, 2004. AUAI Press.

[CF84] J. E. Costa and P. J. Fleisher, editors. *Physical geomorphology of debris flows*, chapter 9, pages 268–317. Springer-Verlag, Berlin, 1984.

[CGOM07] A. Cano, M. Gómez-Olmedo, and S. Moral. Credal nets with probabilities estimated with an extreme imprecise Dirichlet model. In G. de Cooman, I. Vejnarová, and M. Zaffalon, editors, *Proceedings of the Fifth International Symposium on Imprecise Probability: Theories and Applications (ISIPTA '07)*, pages 57–66, Prague, 2007. Action M Agency.

[CHM94] 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.

[CM99] A. Cano and S. Moral. A review of propagation algorithms for imprecise probabilities. In *[dCCMW99]*, pages 51–60, 1999.

[CM02] A. Cano and S. Moral. Using probability trees to compute marginals with imprecise probabilities. *International Journal of Approximate Reasoning*, 29(1):1–46, 2002.

[Coo90] G. F. Cooper. The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence*, 42:393–405, 1990.

[Coz96] F. G. Cozman. Robustness analysis of Bayesian networks with finitely generated convex-sets of distributions. Technical Report CMU-RI-TR 96-41, Robotics Institute, Carnegie Mellon University, 1996.

[Coz00] F. G. Cozman. Credal networks. *Artificial Intelligence*, 120:199–233, 2000.

[Coz05] F. G. Cozman. Graphical models for imprecise probabilities. *International Journal of Approximate Reasoning*, 39(2–3):167–184, 2005.

[CW87] J. H. Costa and G. F. Wiekzorek. *Debris Flows/Avalanches: Process, Recognition and Mitigation*, volume 7. Geol. Soc. Am. Reviews in Engineering Geology, Boulder, CO, 1987.

[DBA06] W.E. Dietrich, D. Bellugi, and R.R.D. Asua. Validation of the shallow landslide model, shalstab, for forest management. In *Land use and watersheds: human influence on hydrology and geomorphology in urban and forest areas edited by M.S. Wigmosta and S.J. Burges*, Washington D.C., 2006. AGU.

[dCC04] 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.

[dCC05] 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.

[dCC07] 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.

[dCCMW99] G. de Cooman, F. G. Cozman, S. Moral, and P. Walley, editors. *ISIPTA '99: Proceedings of the First International Symposium on Imprecise Probabilities and Their Applications*. The Imprecise Probability Project, Universiteit Gent, Belgium, 1999.

[dCZ04] G. de Cooman and M. Zaffalon. Updating beliefs with incomplete observations. *Artificial Intelligence*, 159:75–125, 2004.

[dF74] B. de Finetti. *Theory of Probability*. Wiley, New York, 1974. Two volumes translated from *Teoria Delle probabilità*, published 1970. The second volume appeared under the same title in 1975.

[DFM03] G. Dalla Fontana and L. Marchi. Slope-area relationships and sediment dynamics in two alpine streams. *Hydrological Processes*, 17:73–87, 2003.

[DO06] E. Demircioglu and L. Osadciw. A Bayesian network sensors manager for heterogeneous radar suites. In *IEEE Radar Conference*, Verona, NY, 2006.

[DR03] P. D'Odorico and R. Rigon. Hillslope and channel contributions to the hydrologic response. *Water Resources Research*, 39(5):1–9, 2003.

[dRCdC03] 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.

[Eve79] S. Even. *Graph Algorithms*. Computer Science Press, California, 1979.

[FdRC02] J. C. Ferreira da Rocha and F. G. Cozman. Inference with separately specified sets of probabilities in credal networks. In A. Darwiche and N. Friedman, editors, *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence (UAI 2002)*, pages 430–437, San Francisco, 2002. Morgan Kaufmann.

[FdRC03] J. C. Ferreira da Rocha and F. G. Cozman. Inference in credal networks with branch-and-bound algorithms. In Jean-Marc Bernard, Teddy Seidenfeld, and Marco Zaffalon, editors, *ISIPTA*, volume 18 of *Proceedings in Informatics*, pages 480–493. Carleton Scientific, 2003.

[FZ98] E. Fagiuoli and M. Zaffalon. 2U: an exact interval propagation algorithm for polytrees with binary variables. *Artificial Intelligence*, 106(1):77–107, 1998.

[GH03] P. Grunwald and J. Halpern. Updating probabilities. *Journal of Artificial Intelligence Research*, 19:243–278, 2003.

[GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability; a Guide to the Theory of NP-completeness*. Freeman, San Francisco, 1979.

[Gla05] T. Glade. Linking debris-flow hazard assessments with geomorphology. *Geomorphology*, 66:189–213, 2005.

[GWM04] P. G. Griffiths, R. H. Webb, and T. S. Melis. Frequency and initiation of debris flows in grand canyon, arizona. *Journal of Geophysical Research*, 109:4002–4015, 2004.

[HDVH98] V. Ha, A. Doan, V. Vu, and P. Haddawy. Geometric foundations for interval-based probabilities. *Annals of Mathematics and Artificial Intelligence*, 24(1–4):1–21, 1998.

[IC04] 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.

[IRL97] R. M. Iverson, M. E. Reid, and R. G. Lahusen. Debris-flow mobilization from landslides. *Annual Review of Earth and Planetary Sciences*, 25:85–138, 1997.

[KKW01] U. Kilchenmann, G. Kyburz, and S. Winter. *GEOSTAT user handbook*. Swiss Federal Statistical Office, Neuchatel, Switzerland, 2001.

[Koh03] J. Kohlas. *Information Algebras*. Springer-Verlag, New York, 2003.

[Kun02] R. Kuntner. *A methodological framework towards the formulation of flood runoff generation models suitable in alpine and prealpine regions*. PhD thesis, Swiss Federal Institute of Technology, Zurich, 2002.

[Lev80] I. Levi. *The Enterprise of Knowledge*. MIT Press, London, 1980.

[LR87] R. J. A. Little and D. B. Rubin. *Statistical Analysis with Missing Data*. Wiley, New York, 1987.

[Mai93] D. R. Maidment. *Handbook of Hydrology*. McGraw-Hill, 1993.

[Man03] C. F. Manski. *Partial Identification of Probability Distributions*. Springer-Verlag, New York, 2003.

[MC02] S. Moral and A. Cano. Strong conditional independence for credal sets. *Annals of Mathematics and Artificial Intelligence*, 35(1-4):295–321, 2002.

[MWJ99] 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.

[PA96] I.P. Prosser and B. Abernethy. Predicting the topographic limits of a gully network using a digital terrain model and process threshold. *Water Resources Research*, 32(12):2289–2298, 1996.

[Pap94] C. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[Pea88] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, California, 1988.

[RCP+04] R. Rigon, A. Cozzini, S. Pisoni, G. Bertoldi, and A. Armanini. A new simple method for the determination of the triggering of debris flow. In *10th Congress Interpraevent 2004*, Klagenfurt, 2004. Internationale Forschungsgesellschaft INTERPRAEVENT.

[RIR97] I. Rodriguez-Iturbe and A. Rinaldo. *Fractal river basins - Chance and Self Organization*. Cambridge University Press, Cambridge, UK, 1997.

[RIV79] I. Rodriguez-Iturbe and J.B. Valdes. The geomorphological structure of hydrogeological response. *Water Resources Research*, 15:1409–1420, 1979.

[Ros84] R. Rosso. Nash model relation to horton ratios. *Water Resources Research*, 20(7):914–920, 1984.

[Ser93] USDA Soil Conservation Service. *Hydrology, National Engineering Handbook, Supplement A*. United State Department of Agriculture, Washington D.C., 1993.

[Ser07] AIP Services. *Aeronautical Information Publication Switzerland*. Skyguide, 2007.

[SRH04] A. Skrondal and S. Rabe-Hasketh. *Generalized latent variable modeling: multilevel, longitudinal, and structural equation models*. Chapman and Hall/CRC, Boca Raton, 2004.

[SW93] T. Seidenfeld and L. Wasserman. Dilation for sets of probability. *Annals of Statistics*, 21(3):1139–1154, 1993.

[Tak91] T. Takahashi. *Debris Flow*. A.A. Balkama, Rotterdam, 1991. IAHR Monograph.

[Tes92] B. Tessem. Interval probability propagation. *International Journal of Approximate Reasoning*, 7(3):95–120, 1992.

[VP91] T. Verma and J. Pearl. Equivalence and synthesis of causal models. In P. Bonissone, M. Henrion, L. Kanal, and J. Lemmer, editors, *Proceedings of the Sixth Annual Conference on Uncertainty in Artificial Intelligence (UAI 1990)*, pages 255–270, New York, 1991. Elsevier.

[Wal91] P. Walley. *Statistical Reasoning with Imprecise Probabilities*. Chapman and Hall, New York, 1991.

[Wal96] P. Walley. Inferences from multinomial data: learning about a bag of marbles. *J. R. Statist. Soc. B*, 58(1):3–57, 1996.

[Wel90] M. P. Wellman. Fundamental concepts of qualitative probabilistic networks. *Artificial Intelligence*, 44(3):257–303, 1990.

[Zaf01] M. Zaffalon. Statistical inference of the naive credal classifier. In G. de Cooman, T. Fine, and T. Seidenfeld, editors, *Proceedings of the Second International Symposium on Imprecise Probabilities and Their Applications (ISIPTA '01)*, pages 384–393, The Netherlands, 2001. Shaker Publishing.

[Zaf02] M. Zaffalon. The naive credal classifier. *Journal of Statistical Planning and Inference*, 105(1):5–21, 2002.

[Zaf05] M. Zaffalon. Conservative rules for predictive inference with incomplete data. In F. G. Cozman, R. Nau, and T. Seidenfeld, editors, *ISIPTA '05, Proceedings of the Fourth International Symposium on Imprecise Probabilities and Their Applications*, pages 406–415. SIPTA, 2005.

[ZQP93] N. Zhang, R. Qi, and D. Poole. A computational theory of decision networks. *International Journal of Approximate Reasoning*, 11(2):83–158, 1993.