Bayesian Networks with Imprecise Probabilities: Theory and Application to Classification

G. Corani[1], A. Antonucci[1], and M. Zaffalon[1]

IDSIA, Manno, Switzerland
{giorgio,alessandro,zaffalon}@idsia.ch
www.idsia.ch

Abstract. Bayesian network are powerful probabilistic graphical models for modelling uncertainty. Among others, classification represents an important application: some of the most used classifiers are based on Bayesian networks. Bayesian networks are precise models: exact numeric values should be provided for quantification. This requirement is sometimes too narrow. Sets instead of single distributions can provide a more realistic description in these cases. Bayesian networks can be generalized to cope with sets of distributions. This leads to a novel class of imprecise probabilistic graphical models, called *credal networks*. In particular, classifiers based on Bayesian networks are generalized to so-called *credal classifiers*. Unlike Bayesian classifiers, which always detect a single class as the one maximizing the posterior class probability, a credal classifier may eventually be unable to discriminate a single class. In other words, if the available information is not sufficient, credal classifiers allow for indecision between two or more classes, this providing a less informative but more robust conclusion than Bayesian classifiers.

1 Introduction

Bayesian network [63] are powerful and widespread tools for modelling uncertainty about a domain. These probabilistic graphical models provide a compact and intuitive quantification of uncertain knowledge. After its specification, a Bayesian network can be queried by appropriate inference algorithms in order to extract probabilistic information about the variables of interest. Among others, *classification* represents an important application of Bayesian networks. Some of the most used classifiers proposed within the Bayesian theory of probability, like the *naive Bayes classifier* (Section 8) and the *tree-augmented naive Bayes classifier* (Section 11) can be regarded as learning/inference algorithms for Bayesian networks with particular topologies.

Bayesian networks are *precise* models, in the sense that exact numeric values should be provided as probabilities needed for the model parameters. This requirement is sometimes too narrow. In fact, there are situations where a single probability distribution

cannot properly describe the uncertainty about the state of a variable.[1] In these cases, sets instead of single distributions provide an alternative and more realistic description. E.g., in some cases we may prefer to model our knowledge by interval-valued probabilistic assessments, these corresponding to the specification of the set of distributions compatible with these assessments. Sets of this kind, which are generally required to be closed and convex by some rationality criteria, are called *credal sets* [57]. Approaches where probabilities are quantified in this way are said to be *imprecise* [75].

Bayesian networks can be generalized in order to cope with credal sets. This leads to a novel class of *imprecise* probabilistic graphical models, generalizing Bayesian networks, and called *credal networks* [35]. Expert knowledge is mostly qualitative and it can be therefore naturally described by credal sets instead of single distributions: this makes knowledge-based (or expert) systems represent one of the more natural application of credal networks (e.g., [2, 4, 5]). But even when the focus is on learning probabilities from data, a credal set may offer a more reliable model of the uncertainty, especially when coping with small or incomplete data sets. Thus, classifiers based on Bayesian networks can be profitably extended to become *credal classifiers* based on credal networks. Unlike Bayesian classifiers, which always detect a single class as the one maximizing the posterior class probability[2], a credal classifier works with sets of distributions and may eventually be unable to discriminate a single class as that with highest probability. In other words, if the available information is not sufficient to identify a single class, credal classifiers allow for *indecision* between two or more classes, this representing a less informative but more robust conclusion than Bayesian classifiers.

This chapter describes the main tools of the theory of credal networks in Sections 2–6; it starts by reviewing the general theory of Bayesian network (Section 2) and the fundamental concept of credal set (Section 3), to then illustrate the design and the quantification of the network (Section 4), the query through specific inference algorithms (Section 5), and an environmental application (Section 6). In the second part of the chapter (Sections 7–14) we show how credal networks can be used for classification. In particular, we show how the naive Bayes classifier and the Tree-Augmented Naive (TAN) have been extended to deal with imprecise probabilities, yielding respectively the Naive Credal Classifier (NCC) and the credal TAN (Sections 8–12); this part includes experimental results in texture recognition (Section 9.1) and a discussion of the metrics to evaluate credal classifiers empirically (Section 10). Finally, we review some further credal classifiers (Section 13) and the available software (Section 14).

---

[1] As an example, a condition of *ignorance* about the state of a variable is generally modelled by a uniform distribution, while a more robust model of this ignorance is the whole set of distributions we can specify over this variable.

[2] In the Bayesian framework, the only exception to that is when a condition of *indifference* among two or more classes appears. This corresponds to the situation where the classifier assigns the highest probability to more than a class.

2 Bayesian Networks

We deal with multivariate probabilistic models defined over a collection of variables[3] $\mathbf{X} := \{X_0, X_1, \ldots, X_k\}$. In particular, we consider *graphical* probabilistic models, in the sense that we assume a one-to-one correspondence between the elements of $\mathbf{X}$ and the nodes of a *directed acyclic graph* (DAG) $\mathscr{G}$.[4] Accordingly, in the following we use the terms *node* and *variable* interchangeably. The *parents* of a variable are the variables corresponding to its immediate predecessors according to $\mathscr{G}$. Notation Π_i is used to denote the parents of X_i, for each $X_i \in \mathbf{X}$. Similarly, we define the *children* and, by iterating this relation, the *descendants* of any variable. The graph $\mathscr{G}$ should be intended as a compact description of the conditional independence relations occurring among the variables in $\mathbf{X}$. This is achieved by means of the *Markov condition* for directed graphs: *every variable is independent of its non-descendant non-parents conditional on its parents*. These conditional independence relations can be used to specify a probabilistic model over the whole set of variables $\mathbf{X}$ by means of *local* probabilistic models, involving only smaller subsets of $\mathbf{X}$ (namely, a variable together with its parents, for each submodel). This feature, characterizing directed probabilistic graphical models, will be shared by both the Bayesian networks reviewed here and the *credal networks* introduced in Section 4.

For each $X_i \in \mathbf{X}$, the set of its possible values is denoted as Ω_{X_i}. Here we focus on the case of categorical variables, i.e., we assume $|\Omega_{X_i}| < +\infty$ for each $X_i \in \mathbf{X}$. Similarly, notation Ω_{Π_i} is used for the set of possible values of the joint variable Π_i, corresponding to the parents of X_i. We denote by $P(X_i)$ a probability mass function over X_i, and by $P(x_i)$ the probability that $X_i = x_i$, where x_i is a generic element of Ω_{X_i}. Finally, in the special case of a binary variable X_i, we set $\Omega_{X_i} := \{x_i, \neg x_i\}$, while the (vertical) array notation is used to enumerate the values of a probability mass functions, i.e., $P(X_i) = [\ldots, P(x_i), \ldots]^T$. This formalism is sufficient to introduce the definition of Bayesian network, which is reviewed here below. For a deeper analysis of this topic, we point the reader to Pearl's classical textbook [63].

Definition 1. *A Bayesian network over* $\mathbf{X}$ *is a pair* $\langle \mathscr{G}, \mathbb{P} \rangle$ *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 $\mathscr{G}$ represent probabilistic independence relations between the variables in $\mathbf{X}$. Hence, the conditional probability mass functions associated to the specification of the Bayesian network can be employed to specify a joint mass function $P(\mathbf{X})$ by means of the following factorization formula:

$$P(\mathbf{x}) = \prod_{i=0}^{k} P(x_i | \pi_i),\tag{1}$$

for each $\mathbf{x} \in \Omega_{\mathbf{X}}$, where for each $i = 0, 1, \ldots, k$ the values (x_i, π_i) are those consistent with $\mathbf{x}$.

---

[3] In the sections about classification, the first variable in this collection will be identified with the class and the remaining with the attributes. Notation $\mathbf{X} := (C, A_1, \ldots, A_k)$ will be therefore preferred.

[4] A directed graph is *acyclic* if it does not contains any directed loop.

IV

Bayesian networks provide therefore a specification of a joint probability mass function, describing the probabilistic relations among the whole set of variables. The specification is compact in the sense that only conditional probability mass functions for the variables conditional on (any possible value of) the parents should be assessed. Once a Bayesian network has been specified, a typical task we might consider consists in querying the model to gather probabilistic information about the state of a variable given evidence about the states of some others. This inferential task is called *updating* and it corresponds to the computation of the posterior beliefs about a queried variable X_q, given the available evidence $X_E = x_E$:[5]

$$P(x_q|x_E) = \frac{\sum_{x_M \in \Omega_{X_M}} \prod_{i=0}^{k} P(x_i|\pi_i)}{\sum_{x_M \in \Omega_{X_M}, x_q \in \Omega_{X_q}} \prod_{i=0}^{k} P(x_i|\pi_i)}, \tag{2}$$

where $X_M := \mathbf{X} \setminus (\{X_q\} \cup X_E)$ and the values of x_i and π_i are those consistent with $\mathbf{x} = (x_q, x_M, x_E)$. The variables in X_M are marginalized out of Equation (2) because their values are not available or, in other words, they are *missing*, and this missingness is independent of the actual values of the variables. This represents a special case of the *missing at random* assumption for missing data, which will be discussed in Section 5.3 and Section 9.2.

The evaluation of Equation (2) is an NP-hard task [23], but in the special case of *polytrees*, Pearl's local propagation scheme allows for efficient updating [63]. A polytree is a Bayesian network based on a *singly connected* directed acyclic graph, which is a graph that does not contain any undirected loop.

Bayesian networks are powerful means to model uncertain knowledge in many situations. Yet, the specification of a model of this kind requires the *precise* assessments of the conditional probabilities associated to every variable for any possible value of the parents. Some authors claim this requirement is too strong [75]: an *imprecise* probabilistic evaluation corresponding for instance to an interval and in general to a set of possible estimates would represent a more realistic model of the uncertainty. Thus, we consider a generalization of Bayesian networks in which closed convex sets of probability mass functions instead of single mass functions are provided.

3 Credal Sets

Walley's behavioral theory of *imprecise probabilities* [75] provides a complete probabilistic theory, based on *coherent lower previsions*, that generalizes to imprecision de Finetti's classical theory [43]. A coherent lower prevision can be equivalently expressed by (the lower envelope of) a closed convex set of linear previsions, which are expectations with respect to a finitely additive probability, and hence in one-to-one relationship with mass function 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.

---

[5] A notation with uppercase subscripts (like X_E) is employed to denote vectors (and sets) of variables in $\mathbf{X}$.

3.1 Definition

Following Levi [57], 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 [35] 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 vertices of $K(X)$ is denoted as $\text{ext}[K(X)]$. Enumerating the elements of $\text{ext}[K(X)]$ is then a way to describe a credal set. It is easy to verify that credal sets over binary variables cannot have more than two vertices, while no bounds characterize the possible number of vertices of credal sets over variables with three or more states.

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 $K_{\Omega_X^*}(X)$. The vertices of $K_{\Omega_X^*}(X)$ are the $|\Omega_X^*|$ degenerate mass functions assigning probability one to the single elements of Ω_X^*.
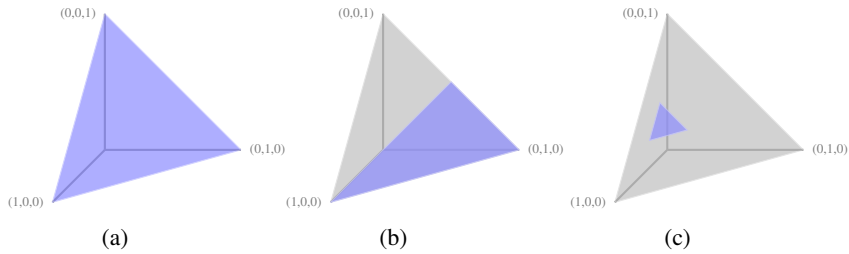

Fig. 1. Geometric representation of credal sets over a ternary variable X (i.e., $\Omega_X = \{x', x'', x'''\}$). The representation is in a three-dimensional space with coordinates $[P(x'), P(x''), P(x''')]^T$. The blue polytopes represent respectively: (a) the vacuous credal set $K_{\Omega_X}(X)$; (b) the credal set defined by constraint $P(x''') > P(x'')$; (c) a credal set $K(X)$ such that $\text{ext}[K(X)] = \{[.1, .3, .6]^T, [.3, .3, .4]^T, [.1, .5, .4]^T\}$.

3.2 Basic Operations with Credal Sets

Given $\tilde{x} \in \Omega_X$, the lower probability for $\tilde{x}$ according to credal set $K(X)$ is

$$\underline{P}^K(\tilde{x}) := \min_{P(X) \in K(X)} P(\tilde{x}). \qquad (3)$$

If there are no ambiguities about the credal set considered in Equation (3), the superscript K is removed and the corresponding lower probability is simply denoted as $\underline{P}(\tilde{x})$. Walley shows that inferences based on a credal set are equivalent to those based only on its vertices [75]. This makes optimization in Equation (3) a combinatorial task. As an example, for the credal set in Figure 1(c), we have $\underline{P}(x') = .1$, $\underline{P}(x'') = .3$ and $\underline{P}(x''') = .4$.

VI

By simply replacing the minimum with the maximum in Equation (3), we can define the upper probability $\overline{P}$. Lower/upper probabilities for any event (including conditional events) in Ω_X can be similarly considered. The *conjugacy*[6] $\overline{P}(\tilde{x}) = 1 - \underline{P}(\Omega_X \setminus \{\tilde{x}\})$ holds, and makes it possible to focus our attention on lower probabilities. Lower/upper expectations can be also considered when coping with generic functions of variable X.

Let us also describe how the basic operations of *marginalization* and *conditioning* can be extended from probability mass functions to credal sets. 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 $P(X)$, obtained marginalizing out Y from $P(X,Y)$, for each $P(X,Y) \in K(X,Y)$. In practical situations, instead of considering all the joint probability mass functions of $K(X,Y)$, marginalization can be obtained by considering only the vertices, and then taking the convex hull, i.e.,

$$K(X) = \mathrm{CH}\left\{ P(X) : P(x) = \sum_{y \in \Omega_Y} P(x,y), \forall x \in \Omega_X, \forall P(X,Y) \in \mathrm{ext}[K(X,Y)] \right\}, \quad (4)$$

where CH denotes the convex hull operator. Concerning *conditioning* with credal sets, we simply perform elements-wise application of Bayes' rule. The conditional credal set is the union of all the conditional mass functions. As in the case of marginalization, the practical computation of a conditional credal set from a joint can be obtained by considering only the vertices of the joint and then taking the convex hull. An expression analogous to that in Equation (4) can be written to compute the conditional credal set $K(X|Y = y)$ from $K(X,Y)$. Note that, in order to apply Bayes' rule, we should assume non-zero probability for the conditioning event $(Y = y)$. This corresponds to having $P(y) > 0$ for each $P(Y) \in K(X)$ (or equivalently for each $P(Y) \in \mathrm{ext}[K(Y)]$), and hence $\underline{P}(y) > 0$. When this condition is not satisfied, other conditioning techniques can be considered. We point the reader to [75, App. J] for a discussion on this issue.

Finally, let us discuss how independence can be intended when knowledge is described by credal sets. In fact, the standard notion of independence (or *stochastic* independence) among two variables X and Y, as adopted within the Bayesian framework, states that X and Y are independent if their joint probability mass function $P(X,Y)$ factorizes, i.e., $P(x,y) = P(x) \cdot P(y)$, for each $x \in \Omega_X$ and $y \in \Omega_Y$. But what should we assume if the knowledge about the two variables is described by a set $K(X,Y)$ instead of a single joint mass function $P(X,Y)$? A possible answer is provided by the notion of *strong independence*: X and Y are strongly independent if they are stochastically independent for each $P(X,Y) \in \mathrm{ext}[K(X,Y)]$. Conditional independence is similarly defined. In the above definition we replace $P(X,Y)$ with $P(X,Y|z)$ and $K(X,Y)$ with $K(X,Y|z)$, and then, if the relation is satisfied for each $z \in \Omega_Z$, we say that X and Y are strongly independent given Z. Strong independence is not the only concept of independence proposed for credal sets. We point the reader to [32] for an overview and [22] for recent developments about other notions of independence in the imprecise-probabilistic framework.

---

[6] We use the same notation for the subsets of the possibility space and the corresponding indicator functions. Accordingly, we can regard set $\Omega_X \setminus \{\tilde{x}\}$ even as function of X returning one when $X \neq \tilde{x}$ and zero otherwise.

3.3 Credal Sets from Probability Intervals

According to the discussion in Section 3.1, a credal set can be specified by an explicit enumeration of its (extreme) probability mass functions. Alternatively, we can consider a set of *probability intervals* over Ω_X:

$$\mathbb{I}_X = \{\mathbb{I}_x : \mathbb{I}_x = [l_x, u_x], 0 \le l_x \le u_x \le 1, x \in \Omega_X\}, \tag{5}$$

The set of intervals can be then used as a set of (linear) constraints to specify the following credal set:

$$K(X) = \left\{ P(X) : P(x) \in \mathbb{I}_x, x \in \Omega_X, \sum_{x \in \Omega_X} P(x) = 1 \right\}. \tag{6}$$

Not all the credal sets can be obtained from a set of probability intervals as in Equation (6), but intervals are often a convenient tool to adopt. $\mathbb{I}_X$ is said to *avoid sure loss* if the corresponding credal set is not empty and to be *coherent* (or *reachable*) if $u_{x'} + \sum_{x \in \Omega_X, x \ne x'} l_x \le 1 \le l_{x'} + \sum_{x \in \Omega_X, x \ne x'} u_x$, 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 [75, 14]. Note that for reachable sets of probability intervals, $\underline{P}(x) = l_x$ and $\overline{P}(x) = u_x$, for each $x \in \Omega_X$. As an example, the credal set in Figure 1(c) is the one corresponding to the reachable set of probability intervals with $\mathbb{I}_{x'} = [.1, .3]$, $\mathbb{I}_{x''} = [.3, .5]$ and $\mathbb{I}_{x'''} = [.4, .6]$. Standard algorithms can compute the vertices of a credal set for which a probability interval has been provided [9]. However, the resulting number of vertices is exponential in the size of the possibility space [71].

3.4 Learning Credal Sets from Data

Probability intervals, and hence credal sets, 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 probability that $X = x$ are given by

$$\mathbb{I}_x = \left[\frac{n(x)}{s + \sum_{x \in \Omega_X} n(x)}, \frac{s + n(x)}{s + \sum_{x \in \Omega_X} n(x)} \right], \tag{7}$$

where $n(x)$ counts the number of instances in the data set in which $X = x$, and s is a hyperparameter that expresses the degree of caution of inferences, usually chosen in the interval $[1, 2]$ (see [76] for details and [10] for a discussion on this choice). To support this interpretation of s, note that if $s = 0$, the credal set associated through Equation (6) to the probability intervals in Equation (7) collapses to a "precise" credal set made of a single extreme point, corresponding to the *maximum likelihood estimator*. On the other side, if $s \to \infty$, the corresponding credal set tends to the vacuous credal set $K_{\Omega_X}(X)$. The probability intervals as in Equation (7) are always reachable. As an example, the credal set in Figure 1(c) can be learned through Equation (7) from a complete dataset about X, with counts $n(x') = 1$, $n(x'') = 3$, $n(x''') = 4$ and $s = 2$. Unlike this example,

there are reachable sets of probability intervals that cannot be regarded as the output of Equation (7) (no matter which are the counts in the data set).

Although in this chapter we only consider the imprecise Dirichlet model, other methods have been also proposed in the literature for learning credal sets from multinomial data (see for instance [20] for an alternative approach and a comparison).

4 Credal Networks

In the previous section we presented credal sets as a more general and expressive model of uncertainty with respect to single probability mass functions. This makes it possible to generalize Bayesian networks to imprecise probabilities. Here we report the basics of the theory for this class of models. We point the reader to [35] for an overview of these models, and to [64] for a tutorial on this topic.

4.1 Credal Network Definition and Strong Extension

The extension of Bayesian networks to deal with imprecision in probability is achieved by means of the notion of credal set. The idea is simple: to replace each conditional probability mass function in Definition 1 with a conditional credal set. This leads to the following definition.

Definition 2. *A credal network over* $\mathbf{X}$ *is a pair* $\langle \mathscr{G}, \mathbb{K} \rangle$*, where* $\mathbb{K}$ *is a set of conditional credal sets* $K(X_i|\pi_i)$*, one for each* $X_i \in \mathbf{X}$ *and* $\pi_i \in \Omega_{\Pi_i}$*.*

In the same way as Bayesian networks specify a (joint) probability mass function over their whole set of variables, credal networks, as introduced in Definition 2, can be used to specify a (joint) credal set over the whole set of variables. According to [35], this corresponds to the *strong extension* $K(\mathbf{X})$ of a credal network, which is defined as the convex hull of the joint mass functions $P(\mathbf{X})$, with, for each $\mathbf{x} \in \Omega_{\mathbf{X}}$:

$$P(\mathbf{x}) = \prod_{i=0}^{k} P(x_i|\pi_i), \quad \begin{array}{l} P(X_i|\pi_i) \in K(X_i|\pi_i), \\ \text{for each } X_i \in \mathbf{X}, \pi_i \in \Pi_i. \end{array} \tag{8}$$

Here $K(X_i|\pi_i)$ can be equivalently replaced by $\text{ext}[K(X_i|\pi_i)]$ according to the following proposition [8].

Proposition 1. *Let* $\{P_j(\mathbf{X})\}_{j=1}^{v} = \text{ext}[K(\mathbf{X})]$*, where* $K(\mathbf{X})$ *is the strong extension of a credal network* $\langle \mathscr{G}, \mathbb{K} \rangle$ *are joint mass functions obtained by the product of vertices of the conditional credal sets, i.e., for each* $\mathbf{x} \in \Omega_{\mathbf{X}}$*:*

$$P_j(\mathbf{x}) = \prod_{i=0}^{k} P_j(x_i|\pi_i), \tag{9}$$

for each $j = 1, \ldots, v$*, where, for each* $i = 0, \ldots, k$ *and* $\pi_i \in \Omega_{\Pi_i}$*,* $P_j(X_i|\pi_i) \in \text{ext}[K(X_i|\pi_i)]$*.*

According to Proposition 1, we have that the vertices of the strong extension of a credal network can be obtained by combining the vertices of the conditional credal sets involved in the definition of credal network. Note that this makes the number of vertices of the strong extension exponential in the input size.

Example 1 (A simple credal network). Consider a credal network associated to the graph in Figure 2. According to Definition 2, the specification requires the assessment of the (unconditional) credal set $K(X_0)$, and two conditional credal sets (one for each value of parent X_0) for X_1 and X_2. Note also that, according to Proposition 1, the vertices of the strong extension $K(X_0, X_1, X_2)$ cannot be more than 2^5.
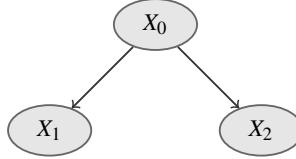

Fig. 2. A credal network over three binary variables. Concerning quantification, we set $\text{ext}[K(X_0)] = \{[.2,.8]^T,[.5,.5]^T\}$, $\text{ext}[K(X_1|x_0)] = \{[.3,.7]^T,[.4,.6]^T\}$, $\text{ext}[K(X_1|\neg x_0)] = \{[.1,.9]^T,[.2,.8]^T\}$, $\text{ext}[K(X_2|x_0)] = \{[.5,.5]^T,[.6,.4]^T\}$, $\text{ext}[K(X_2|\neg x_0)] = \{[.7,.3]^T,[.8,.2]^T\}$.

The key for the decomposition, as in Equation (1), of the joint probability mass function associated to a Bayesian network are the *stochastic* conditional independence relations outlined by the graph underlying the network according to the Markov condition. Similarly, the decomposition characterizing the strong extension of a credal network follows from the *strong* conditional independence relations associated to the graph. Other joint credal sets, alternative to the strong extension, might correspond to different notions of independence adopted in the semantic of the Markov condition. We point the reader to [21], for an example of credal networks based on a different notion of independence.

4.2 Non-Separately Specified Credal Networks

In the definition of strong extension as reported in Equation (8), each conditional probability mass function is free to vary in its conditional credal set independently of the others. In order to emphasize this feature, credal networks of this kind are said to be with *separately specified credal sets*, or simply separately specified credal networks.

Separately specified credal networks are the most commonly used type of credal network, but it is possible to consider credal networks whose strong extension cannot be formulated as in Equation (8). 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 assigned to some other conditional mass functions. A credal network of this kind is called *non-separately specified*.

Some authors considered so-called *extensive* specifications of credal networks [66], where instead of a separate specification for each conditional mass function associated to X_i, 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. This corresponds to assume constraint between the specification of the conditional credal sets $K(X_i|\pi_i)$ for the different values of $\pi_i \in \Omega_{\Pi_i}$. The

strong extension of an extensive credal network is obtained as in Equation (8), 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 (and then taking the convex hull).

Example 2 (Extensive specification of a credal network). Consider the credal network defined in Example 1 over the graph in Figure 2. Keep the same specification of the conditional credal sets, but this time use the following (extensive) constraints: when the first vertex of $K(X_1|x_0)$ is chosen, the first vertex of $K(X_1|\neg x_0)$ has to be chosen too; similarly for the second vertex of $K(X_1|x_0)$ and for variable X_2. This corresponds to assume the following possible values for the conditional probability tables:

$$P(X_1|X_0) \in \left\{ \begin{bmatrix} .3 & .1 \\ .7 & .9 \end{bmatrix}, \begin{bmatrix} .4 & .2 \\ .6 & .8 \end{bmatrix} \right\} \quad P(X_2|X_0) \in \left\{ \begin{bmatrix} .5 & .7 \\ .5 & .3 \end{bmatrix}, \begin{bmatrix} .6 & .8 \\ .4 & .2 \end{bmatrix} \right\}. \quad (10)$$

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

Example 3 (Learning from incomplete data). Given three binary variables X_0, X_1 and X_2 associated to the graph in Figure 3, we want to learn the model probabilities from the incomplete data set in Table 1, assuming no information about the process making the observation of X_1 missing in the last instance of the data set. A possible approach is to learn two distinct probabilities from the two complete data sets corresponding to the possible values of the missing observation,[7] and use them to specify the vertices of the conditional credal sets of a credal network.
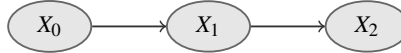

Fig. 3. The graph considered in Example 3.

X_0	X_1	X_2
x_0	x_1	x_2
$\neg x_0$	$\neg x_1$	x_2
x_0	x_1	$\neg x_2$
x_0	$*$	x_2

Table 1. A data set about three binary variables; "$*$" denotes a missing observation.

---

[7] The rationale of considering alternative complete data sets in order to conservatively deal with missing data will be better detailed in Section 5.3.

To make things simple we compute the probabilities for the joint states by means of the relative frequencies in the complete data sets. Let $P_1(X_0, X_1, X_2)$ and $P_2(X_0, X_1, X_2)$ be the joint mass functions obtained in this way, which define the same conditional mass functions for

$$P_1(x_0) = P_2(x_0) = \frac{3}{4}$$
$$P_1(x_1|\neg x_0) = P_2(x_1|\neg x_0) = 0$$
$$P_1(x_2|\neg x_1) = P_2(x_2|\neg x_1) = 1;$$

and different conditional mass functions for

$$
\begin{array}{ll}
P_1(x_1|x_0) = 1 & P_2(x_1|x_0) = \frac{2}{3} \\
P_1(x_2|x_1) = \frac{2}{3} & P_2(x_2|x_1) = \frac{1}{2}.
\end{array}
\tag{11}
$$

We have therefore obtained two, partially distinct, Bayesian network specifications over the graph in Figure 3. The conditional probability mass functions of these networks are the vertices of the conditional credal sets for the credal network we consider. Such a credal network is non-separately specified. To see that, just note that if the credal network would be separately specified the values $P(x_1|x_0) = 1$ and $P(x_2|x_1) = \frac{1}{2}$ could be regarded as a possible instantiation of the conditional probabilities, despite the fact that there are no complete data sets leading to this combination of values.

Although their importance in modelling different problems, non-separate credal networks have received relatively little attention in the literature. Most of the algorithms for credal networks inference are in fact designed for separately specified credal networks. However, two important exceptions are two credal classifiers which we present later: the naive credal classifier (Section 9) and the credal TAN (Section 12).

Furthermore, in a recent work [8] it has been shown that non-separate credal networks can be equivalently described as separate credal networks augmented by a number of auxiliary parents nodes enumerating only the possible combinations for the constrained specifications of the conditional credal sets. This can be described by means of the two following examples.

Example 4 (Separate specification of non-separate credal networks). Consider the extensive credal network in Example 2. Nodes X_1 and X_2 are characterized by an extensive specification. Thus we add to the model two auxiliary variables X_3 and X_4, that become parents of X_1 and X_2 respectively. The resulting graph is that in Figure 4(a). Each auxiliary node should index the tables in the specification of its children. In Equation (10) we have two tables for each node. Thus, we assume nodes X_3 and X_4 to be binary, and we redefine the following quantification for nodes X_1 and X_2: $P(X_1|X_0, x_3) = P_1(X_1|X_3)$ and $P(X_1|X_0, \neg x_3) = P_2(X_1|X_3)$, where P_1 and P_2 are the two tables in the specification. We similarly proceed for X_2. Finally, regarding nodes X_2 and X_3, we set a vacuous specification, i.e., $K(X_2) := K_{\Omega_{X_2}}(X_2)$ and similarly for X_3. Note that this credal network is separately specified. Let $K(X_0, X_1, X_2, X_3, X_4)$ denote the strong extension of this network, and $K(X_0, X_1, X_2)$ the joint credal set obtained by marginalizing out X_3 and X_4. The result in [8] states that $K(X_0, X_1, X_2)$ coincides with the strong extension of the extensive credal network of Example 2.

We similarly proceed for the credal network in Example 3. The constraints between P_1 and P_2 in Equation (11) correspond to a non-separate specification of the values

of the conditional probabilities of X_1 and X_2. As in the previous case, we add to the graph in Figure 3 an auxiliary node X_3, which is a parent of both X_1 and X_2, and for the quantification we proceed as in the previous example. This leads to the credal network in Figure 4 (b).

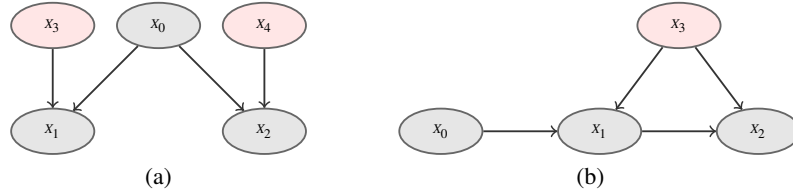(a)   (b)

Fig. 4. Modelling non-separately specific conditional credal sets with control nodes (in pink).

This procedure can be easily applied to any non-separate specification of a credal network. We point the reader to [8] for details.

5   Computing with Credal Networks

5.1   Credal Networks Updating

By perfect analogy with what we have done for Bayesian networks in Section 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 of the posterior probability $\underline{P}(x_q|x_E)$ with respect to the network strong extension $K(\mathbf{X})$. Equation (2) generalizes as follows:

$$\underline{P}(x_q|x_E) = \min_{j=1,\dots,v} \frac{\sum_{x_M} \prod_{i=0}^{k} P_j(x_i|\pi_i)}{\sum_{x_M,x_q} \prod_{i=0}^{k} P_j(x_i|\pi_i)}, \tag{12}$$

where $\{P_j(\mathbf{X})\}_{j=1}^{v}$ are the vertices of the strong extension. A similar expression with a maximum replacing the minimum defines upper probabilities $\overline{P}(x_q|x_E)$. Note that, according to Proposition 1, for separately specified credal networks, the number v of vertices of the strong extension is exponential in the input size. Thus, Equation (12) cannot be solved 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[8], and NP$^{\text{PP}}$-complete for general credal networks [37]. Nevertheless, a number of exact and approximate algorithm for credal networks updating has been developed. A summary about the state of the art in this field is reported in Section 5.2.

---

[8] We extend to credal networks the notion of polytree introduced for Bayesian networks in Section 2.

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

$$\Omega_{X_q}^* := \left\{ x_q \in \Omega_{X_q} : \nexists x_q' \in \Omega_{X_q} s.t. \overline{P}(x_q|x_E) < \underline{P}(x_q'|x_E) \right\}. \tag{13}$$

Criteria other than interval dominance have been proposed in the literature and formalized in the more general framework of decision making with imprecise probabilities [72]. Most of these criteria require the availability of the *posterior credal set*:

$$K(X_q|x_E) = \text{CH}\left\{ P_j(X_q|x_E) \right\}_{j=1}^v. \tag{14}$$

As an example, the set of non-dominated outcomes $\Omega_{X_q}^{**}$ according to the *maximality* criterion [75] is obtained by rejecting the outcomes whose probabilities are dominated by those of some other outcome, for any distribution in the posterior credal set in Equation (14), i.e.,

$$\Omega_{X_q}^{**} := \left\{ x_q \in \Omega_{X_q} : \nexists x_q' \in \Omega_{X_q} s.t. P(x_q|x_E) < P(x_q'|x_E) \forall P(X_q|x_E) \in \text{ext}[K(X_q|x_E)] \right\}. \tag{15}$$

Maximality is more informative than interval dominance, i.e., $\Omega_{X_q}^{**} \subseteq \Omega_{X_q}^*$. Yet, most of the algorithms for credal networks only returns the posterior probabilities as in Equation (12), while the posterior credal set as in Equation (14) is needed by maximality. Notable exceptions are the models considered in Section 9 and Section 12, for which the computation of the set as in Equation (15) can be performed without explicit evaluation of the posterior credal set. In other cases, a procedure to obtain an (outer) approximation of the credal set in Equation (14) can be used [3].

5.2 Algorithms for Credal Networks Updating

Despite the hardness of the problem, a number of algorithms for exact updating of credal networks have been proposed. Most of these methods generalize existing techniques for Bayesian networks. Regarding Pearl's algorithm for efficient updating on polytree-shaped Bayesian networks [63], a direct extension to credal networks is not possible. 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 credal networks, because $\underline{P}(X_q|x_E)$ and $\overline{P}(X_q|x_E)$ are not normalized in general. A remarkable exception 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.

Another approach to exact inference is based on a generalization of the *variable elimination* techniques for Bayesian networks. In the credal case, this corresponds to a *symbolic* variable elimination, where each elimination step defines a *multilinear* constraint among the different conditional probabilities where the variable to be eliminated appears. Overall, this corresponds to a mapping between credal networks updating and *multilinear programming* [14]. Similarly, a mapping with an integer linear programming problem can be achieved [13]. Other exact inference algorithms examine potential vertices of the strong extension according to different strategies in order to produce the required lower/upper values [15, 35, 66, 67].

Concerning approximate inference, *loopy propagation* is a popular technique that applies Pearl's propagation to multiply connected Bayesian networks [61]: propagation is iterated until probabilities converge or for a fixed number of iterations. In [53], Ide and Cozman extend these ideas to belief updating on credal networks, by developing a loopy variant of 2U that makes the algorithm usable for multiply connected binary credal networks. This idea has further exploited by the *generalized loopy 2U*, which transforms a generic credal network into an equivalent binary credal network, which is indeed updated by the loopy version of 2U [6]. Other 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 [18, 68, 49, 71], while the latter produce intervals that are enclosed by the correct probability interval [15, 34]. 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). Overviews of inference algorithms for imprecise probabilities have been published by Cano and Moral (e.g., [17]).

5.3 Modelling and Updating with Missing Data

In the updating problem described in Equation (12), the evidence x_E is assumed to report the actual values of the variables in X_E. This implicitly requires the possibility of making *perfectly reliable* observations. Clearly, this is not always realistic. An example is the case of *missing data*: we perform an observation but the outcome of the observation is not available. 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 (MAR, [58]). This allows missing data to be neglected, thus turning the incomplete data problem into one of complete data. In particular, MAR implies that the probability of a certain value to be missing does not depend on the value itself, neither on other non-observed values. For instance, the temporary breakdown of a sensor produces MAR missing data, because the probability of missing is one, regardless of the actual value[9]. As a further example, consider a medical center where test B is performed only if test

---

[9] In this case, the data are *missing completely at random* (MCAR), which is a special case of MAR [54]

A is positive; the missingness of B is MAR because its probability to be missing only depends on the observed value of A. Yet, MAR is not realistic in many cases. Consider for instance an exit poll performed during elections, where the voters of the right-wing party sometimes refuse to answer; in this case, the probability of an answer to be missing depends on its value and thus the missingness is non-MAR. Ignoring missing data that are non-MAR can lead to unreliable conclusions; in the above example, it would underestimate the proportion of right-wing voters. However, it is usually not possible to test MAR on the incomplete observations; if MAR does not appear tenable, more conservative approaches than simply ignoring missing data are necessary in order to avoid misleading conclusions.

De Cooman and Zaffalon have developed an inference rule based on much weaker assumptions than MAR, which deals with near-ignorance about the missingness process [39]. This result has been extended by Zaffalon and Miranda [82] 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: [10]

$$K(X_q||^{X_I}x_E) := \mathrm{CH}\left\{P_j(X_q|x_E,x_I)\right\}_{x_I \in \Omega_{X_I}, j=1,\dots,v},\qquad(16)$$

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

When coping only with the missing-at-random variables (i.e., if X_I is empty), Equation (16) becomes a standard updating task to be solved by some of the algorithms in Section 5.2. Although these algorithms cannot be applied to solve Equation (16) if X_I is not empty, a procedure to map a conservative inference task as in Equation (16) into a standard updating task as in Equation (12) over a credal network defined over a wider domain has been developed [7]. The transformation is particularly simple and consists in the augmentation of the original credal network with an auxiliary child for each non-missing-at-random variable, with an extensive quantification. This procedure is described by the following example.

Example 5 (CIR-based updating by standard updating algorithms). Consider the credal network in Example 1. Assume that you want to update your beliefs about X_0, after the observation of both X_1 and X_2. The observation of X_1 is x_1, while the outcome of the

---

[10] This updating rule can be applied also to the case of *incomplete* observations, where the outcome of the observation of X_I is missing according to a non-missing-at-random process, but after the observation some of the possible outcomes can be excluded. If $\Omega'_{X_I} \subset \Omega_{X_I}$ is the set of the remaining outcomes, we simply rewrite Equation (16), with Ω'_{X_I} instead of Ω_{X_I}.

observation of X_2 is missing, and the MAR assumption seems not tenable. Accordingly we update the model by means of conservative inference rule as in Equation (16) to compute $K(X_0||^{X_2}x_1)$. In order to map this CIR-based updating task into a standard updating, let us perform the following transformation. As described in Figure 5, we first augment the network with an auxiliary binary variable X_3, which is a child of X_2. Then we extensively quantify the relation between these two nodes as:

$$P(X_3|X_2) \in \left\{ \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right\}, \tag{17}$$

which can be indeed formulated as a separate specification by augmenting the network with a binary node X_4, which is a parent of X_3, according to the procedure described in Example 4. The result in [7] states that $K(X_0||^{X_2}x_1) = K(X_0|x_1, x_3)$, where x_3 is the state corresponding to the first row of the tables in Equation (17). The lower and upper probabilities associated to the posterior credal set can be therefore computed by standard updating.
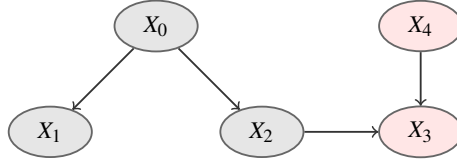

Fig. 5. Modelling non-missing-at-random observation of X_2 in credal network.

6 An Application: Assessing Environmental Risk by Credal Networks

In the previous sections we gave the reader a number of theoretical tools for both modelling and interacting with credal networks. In this section, we want to present a real-world application of these methods consisting in a specific risk analysis task.[11] The credal network merges into a single coherent framework different kinds of domain knowledge: deterministic equations, human expertise, and historical data are used to quantify the network in its different parts. After the model specification, risk analysis can be automatically performed by means of some of the updating algorithms in Section 5.2.

6.1 Debris Flows

Debris flows are among the most dangerous and destructive natural hazards that affect human life, buildings, and infrastructures (see Figure 6). They are gravity-induced mass

---

[11] We point the reader to [64] for a gentle introduction to the issues related to the practical implementation of a credal network in knowledge-based expert systems.

movements intermediate between landslides and water floods. The flow is composed of a mixture of water and sediment with a characteristic mechanical behavior varying with water and soil content. According to [31], 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 snow-melt), and sparse vegetation. As mentioned in [48], several hypotheses have been formulated to explain mobilization of debris flows and this aspect still represents a research field. According to the model proposed by [70], the mechanism to disperse the materials in flow depends on the properties of the materials (like *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. In fact, the analysis of historical data and the role of human expertise are still fundamental for hazard identification as many aspects of the whole process are still poorly understood. For these reasons, a credal network seems to be a particularly suitable model for approaching a problem of this kind.

Fig. 6. Debris flows examples.

6.2   The Credal Network

In order to implement a credal network able to estimate the level of risk of a debris flow happening in a particular area, we first define the *Movable Debris Thickness* as the

depth of debris likely to be transported downstream during a flood event. Such variable represents an integral indicator of the hazard level. Then we identify a number of triggering factors which may affect the value of this thickness. Once we have identified the factors, the specification of the directed acyclic graph associated to these variables can be achieved assuming a *causal* interpretation to the arcs of the graph.[12] Figure 7 depicts the resulting graph. We point the reader to [5] for a detailed description of the different variables in this model. Here let us only report the information we need to understand the key features of both the modelling and the inference with a model of this kind.
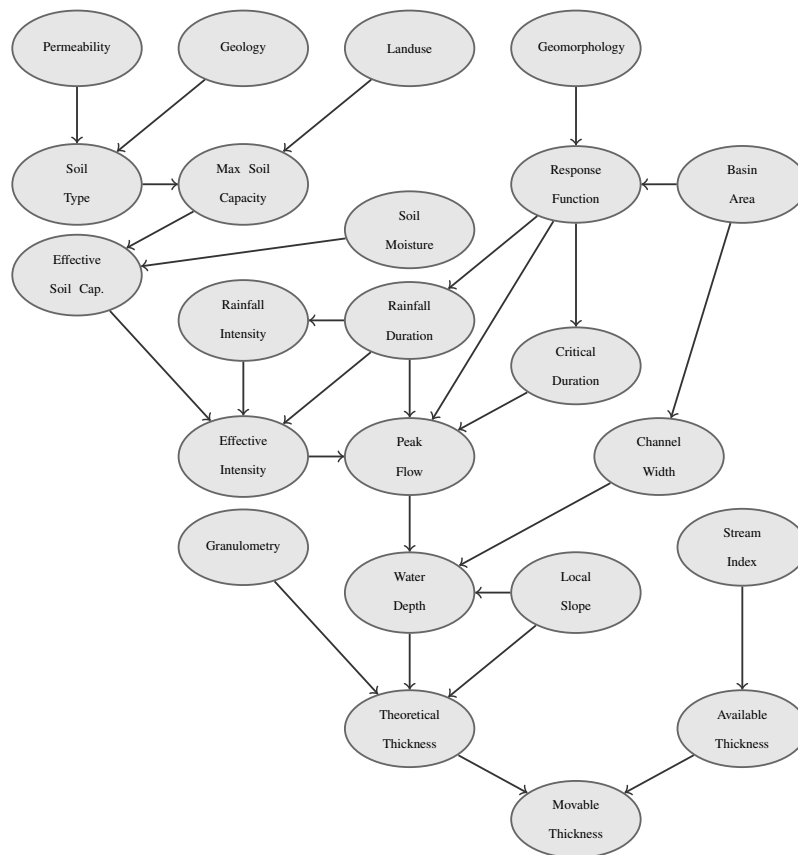

Fig. 7. A credal network for environmental risk analysis.

---

[12] Remember that, according to the Markov condition, the directed graph is a model of conditional independence relations. The causal interpretation is therefore not always justified.

Credal networks have been defined only for categorical variables.[13] Some of the variables in the network are natively categorical. This is for instance the case of variable *Land Use*, whose six possible values are: *Forest*, *Pasture*, *Rivers and water bodies*, *Improductive vegetation*, *Bare soils and rocks*, *Edificated surfaces*. Some other variables, like for example the *Movable Debris Thickness* are numerical and continuous. A discretization like that in Table 2 is therefore required.

Range	Risk Level	Symbol
< 10 cm	low risk	$<$
$10 - 50$ cm	medium risk	$=$
> 50 cm	high risk	$>$

Table 2. Discretization of variable *Movable Debris Thickness* and corresponding interpretation in terms of actual level of risk. The same discretization has been used also for variables *Theoretical Thickness* and *Available Thickness*.

Regarding the probabilistic quantification of the conditional values of the variables given the parents, as we have based our modelling on a geomorphological model of the triggering of the debris flow, we have deterministic equations for most of the variables in the network. Given an equation returning the numerical value of a child given the values of the parents, we can naturally induce the quantification of the corresponding conditional probabilities. A quantification of this kind is clearly precise (i.e., described by a single conditional probability table) and in particular deterministic (i.e., the columns corresponding to the different conditional mass functions assign all the mass to a single outcome and zero to the others). As an example, the *Movable Debris Thickness* is the minimum between the *Theoretical Thickness* and the *Available Thickness* and this corresponds to the following conditional probability table (where the symbols in Table 2 are used to denote the states of the variables):

Theoretical		$< = >$	$< = >$	$< = >$
Available		$< < <$	$= = =$	$> > >$
	$<$	1 1 1	1 0 0	1 0 0
Movable	$=$	0 0 0	0 1 1	0 1 0
	$>$	0 0 0	0 0 0	0 0 1

For some other variables, including also all the root (i.e., parentless) nodes, we have not relations of this kind. In this cases, we used, when available, historical dataset, from which we obtained conditional credal sets by means of the imprecise Dirichlet model as in Equation (7). Note that, especially in the conditional case, the amount of data can be relatively small, and the difference between the credal sets we learn by means of the imprecise Dirichlet model and the precise is not trivial. This is a further justification for our choice of modelling the problem by means of a credal instead of a Bayesian network.

---

[13] In a certain sense, the work in [11] can be implicitly regarded as an exception of this statement. Yet, research on credal network with continuous variable is still in its early stage.

Finally, the counterpart of the role of human expertise in the evaluation is the fact that some of the credal set we quantify in our model cannot be obtained from data neither from deterministic relations. In these cases, we ask an expert to report his knowledge. Notably, the possibility of expressing his beliefs by intervals of probability instead of single values makes the description much more realistic.[14] Overall, we achieve in this way the quantification of a credal network over the directed acyclic graph in Figure 7.

The practical application of a model of this kind consists in the computation of the posterior probability intervals for the three different level of the risk given the observed values for some of the other variables in the network for the particular scenarion under consideration. The updating has been provided by means of the algorithm in [14]. The histograms in Figure 10 report the posterior intervals obtained for three different scenarios. Note that, according to the interval-dominance criterion in the scenario (a) we can reject the second and the third histogram, and conclude that a level of high risk occurs. Regarding (b), the high risk dominates the low risk, which is therefore rejected, but there is an indecision between high and medium risk, while in the scenario (c) no dominance is present and we are in a situation of complete indecision between the three different levels of risk. This kind of analysis can be automatically performed by the credal network on extensive areas, this providing an importanting support to the experts for this problem. Figure 9 reports an extensive analysis for the basin in Figure 8.
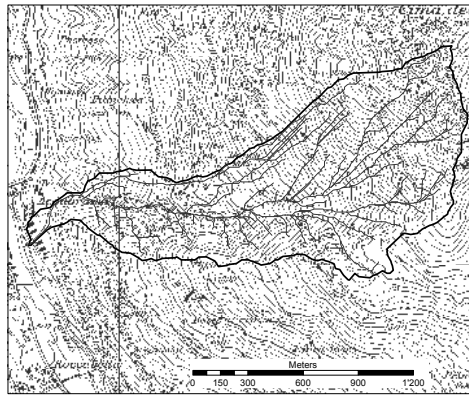

Fig. 8. Acquarossa Creek Basin (area 1.6Km^2, length 3.1Km).

---

[14] We thanks Dott. Andrea Salvetti, the environmental expert who was involved in this quantification task and in many other aspects of this project.
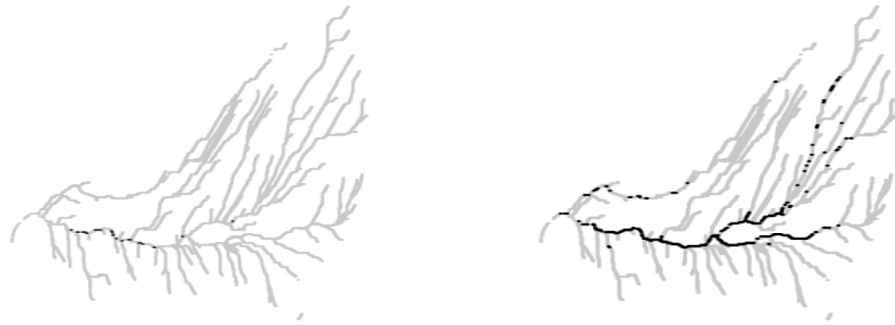
Fig. 9. Spatially distributed identifications for the basin in Figure 8 and rainfall return periods of 10 (left) and 100 (right) years. The points for which the credal network predicts the lower class of risk are depicted in gray, while black refers to points where higher levels of risk cannot be excluded.
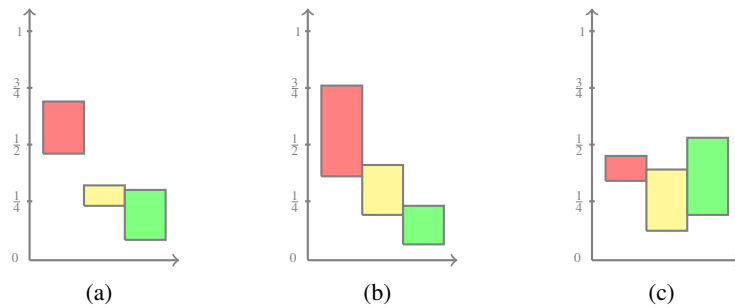

Fig. 10. Posterior probability intervals for the three level of risk (colors red, yellow and green correspond respectively to high, medium and low risk).

7 Credal Classifiers

In the rest of this chapter we show how credal networks can be used to deal with a classical field of data mining, namely *classification*. Classification is the problem of predicting the *class* of a given object, on the basis of some attributes (*features*) of it. A historical example is the iris problem designed by Fisher in 1936: the goal is to predict the species of Iris (among three possible categories) on the basis of four features, namely the length and the width of the sepal and the petal.

Training a probabilistic classifier corresponds to estimating from data the *joint* distribution $P(C, \mathbf{A})$, where C denotes the class variable and $\mathbf{A} = \{A_1, \ldots, A_k\}$ the set of k features. In the Bayesian framework, the estimation of the joint distribution starts by initializing it to an initial value (the *prior*), which represents the beliefs of the investigator *before* analyzing the data; the prior thus enables to model domain knowledge. Then the *likelihood* function is computed from the data, modelling the evidence coming from the observations. Prior and likelihood are multiplied, leading to a *posterior* joint distribution. As described in Section 2, when dealing with Bayesian networks, one does not need to specify the full joint; it is enough to specify the local conditional distributions, and the network automatically represents the joint. A trained classifier is assessed by checking its accuracy at classifying instances. To classify an instance characterized by the assignment $\mathbf{a} = \{a_1, \ldots, a_k\}$ of the features, the conditional distribution $P(C|\mathbf{a})$ is computed from the posterior joint.

A traditional criticism of Bayesian methods is the need for specifying a prior distribution. In fact, prior information is generally difficult to quantify; moreover one often prefers to let the data speak by themselves, without introducing possibly subjective prior beliefs. As for classification in particular, Bayesian classifiers might happen to return *prior-dependent* classifications, i.e., the most probable class varies under different priors. As the choice of any single prior entails some arbitrariness, prior-dependent classifications are typically unreliable: in fact, they translate the arbitrariness of the choice of the prior into arbitrariness of the conclusions. Prior-dependent classifications are more frequent on small data sets; on large data sets, classifications are less sensitive on the choice of the prior. Nevertheless, as shown in Section 9.1, unreliable prior-dependent classifications can be present also in large data sets. Most often, one deals with the choice of the prior by setting a *uniform* prior, because it looks non-informative; yet, such an approach has important drawbacks, as shown in the following example, inspired to [78].

Let us consider a bag containing blue marbles and red marbles; no drawings have been made from the urn. Which is the probability of getting a red (or blue) marble in the next draw? Using the uniform prior, one should assign the same probability 0.5 to both colors. This underlies the (very strong) assumption that the urn contains an equal number of red and blue marbles; in the subjective interpretation of probability, this means that one is equally available to bet an amount of money 0.5 on either red or blue, in a gamble with reward 1 and 0 for respectively a correct and a wrong prediction. In fact, the uniform prior is a model of *prior indifference*. However, we are ignorant about the content of the urn rather than indifferent between the two colors; in this condition, the only reliable statement is that the proportion of red (or blue) marbles is comprised between 0 and 1. Walley's 'theory of imprecise probability [77] states that such *prior*

ignorance should be represented by a *set* of prior distributions rather than by a single prior. The adoption of a set of priors (letting the proportion of blue and red vary between 0 and 1) prevents betting on any of the two colors, which is more sensible, under ignorance, than being equally available to bet on both.

Credal classifiers extend Bayesian classifiers to imprecise probabilities; they represent *prior-ignorance*[15] by specifying a (credal) set of priors, often using the IDM [78]. The credal set of the IDM is then turned into a set of posterior by element-wise application of Bayes' rule: in fact, training a credal classifier corresponds to update the set of priors with the likelihood, yielding a *set* of posteriors. Credal classifiers detect prior-dependent instances by checking whether the most probable class is consistent or not across the set of posteriors. If the instance is prior-dependent, a credal classifier returns a set of classes, drawing a less informative but more robust conclusion than a Bayesian classifier.

However, besides prior-ignorance, there is another kind of ignorance involved in the process of learning from data, i.e., ignorance about the missingness process (MP). Usually, classifiers ignore missing data, assuming missing data to be MAR. In general there is no way to verify the MAR assumption on the incomplete data; furthermore assuming MAR when it does not hold can cause to a large decrease of accuracy [65]. However, credal classifiers have been also extended to conservatively deal with non-MAR missing data [29], relying on CIR, namely by considering all the data sets consistent with the observed incomplete data set.

Other classifiers, besides the credal ones, suspend the judgment on doubtful instances. For instance, a *rejection rule* can be set on any classifier, refusing to classify the instances (and hence returning the whole set of classes), where the probability of the most probable class is below a certain threshold. A more sophisticated approach has been developed by del Coz et al. [33]: their algorithm determines which set of classes to return (possibly a single class), on the basis of the posterior distribution computed by the classifier; this algorithm can be therefore applied to any probabilistic classifier. The returned set of classes is identified in order to maximize the F-measure of the issued classifications.

However, both the rejection rule and the algorithm of [33] work on a single probability distribution; instead, credal classifiers deal with a *set* of posterior distributions. The practical difference can be appreciated by considering a classifier trained on a very small learning set. The credal classifier will be very likely to suspend the judgment on any new instance, as most instances are going to be prior-dependent. On the contrary, a traditional classifier equipped with the rejection rule or with the algorithm of [33] blindly trusts the computed posterior distribution, without considering that for instance on small data sets it might be largely sensitive on the choice of the prior.

8 Naive Bayes

The naive Bayes classifier (NBC) [40] "naively" assumes the features $A_1, \ldots, A_k$ to be independent given the class C. According to the Markov condition introduced in

---

[15] More precisely, prior *near-ignorance*; full ignorance is not compatible with learning, as shown in Section 7.3.7 of Walley [77].
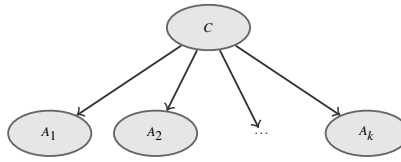
Fig. 11. The naive Bayes classifier.

Section 2, these conditional independence relations can be graphically depicted by the directed graph in Figure 11. These assumptions introduce a severe bias in the estimate of probabilities, as the real data generation mechanism does *not* generally satisfy such condition. As a consequence of such unrealistic assumption, NBC is often overconfident in its predictions, assigning a very high probability to the most probable class [50]; this phenomenon is emphasized if redundant features are present.

Despite the simplistic naive assumption, NBC performs surprisingly well under 0-1 loss[16][40, 50]. A first reason is that the bias of the probability estimates may not matter under 0-1 loss: given two classes c' and c'' (c' being the correct one), even a severe bias in the estimate of $P(c')$ will not matter, provided that $P(c') > P(c'')$ [46]. The good performance of NBC can be further explained by decomposing the misclassification error into bias and variance [46]: NBC has indeed high bias, but this problem is often successfully remediated by low variance. Especially on small data sets, low variance is more important than low bias; in this way, NBC can outperform more complex classifiers. Instead, more parameterized classifiers tend to outperform NBC on large data sets. The low variance of NBC is due to the low number of parameters, which is a consequence of the naive assumption, which prevents modelling correlations between features. For instance, in comparison with C4.5 (which has lower bias but higher variance), NBC is generally more accurate on smaller sample sizes, but generally outperformed on larger data sets [55].

A further factor which contributes to the good performance of NBC is feature selection, which typically removes the most correlated features and thus makes the naive assumption more realistic. In fact, NBC can be even very competitive, when trained with a carefully designed feature set. For instance, the CoIL challenge [74] was won by a NBC entry [41], which outperformed more complicated models such as SVMs or neural networks. The data set of the competition was characterized by several correlated features and noisy data; a later analysis of the contest [74] showed that variance was a much bigger problem than bias for this data set. However, key factors for the success of NBC were feature selection and the introduction of a particular feature, obtained by taking the Cartesian product of two important features, which enabled NBC to account for the interactions between such two features. In [44], a criterion aimed at removing irrelevant or redundant features on the basis of conditional mutual information is designed; under this feature selection, NBC is competitive with SVMs and boosting in several problems.

---

[16] This loss function is also known as *misclassification error*: if the most probable class is the correct one the loss is zero and one otherwise.

Further strengths of NBC are computational speed and easy handling of missing data, at least under the MAR assumption. However, if MAR is not assumed, the computation becomes quite complicated; see for instance the algorithms designed in [65].

NBC has been recognized as one of the ten most influential data mining algorithms [80] and in fact there have been also countless NBC variants designed to improve its performance; comprehensive references can be found for instance in [50] and [51].

8.1 Mathematical Derivation

Let us denote by C the classification variable (taking values in Ω_C) and as $A_1,\ldots,A_k$ the k feature variables (taking values from the finite sets $\Omega_{A_1},\ldots,\Omega_{A_k}$).

We denote by $\theta_{c,\mathbf{a}}$ the chance (i.e., the unknown probability about which we want to make inference) that $(C,A_1,\ldots,A_k)=(c,\mathbf{a})$, by $\theta_{a_i|c}$ the chance that $A_i=a_i$ given that $C=c$, by $\theta_{\mathbf{a}|c}$ the chance that $A_1,\ldots,A_k=(a_1,\ldots,a_k)$ conditional on c.

The naive assumption of independence of the features given the class can be expressed as:

$$\theta_{\mathbf{a}|c} = \prod_{i=1}^{k}\theta_{a_i|c}. \tag{18}$$

We denote by $n(c)$ and $n(a_i,c)$ the observed counts of $C=c$ and of $(A_i,C)=(a_i,c)$; by $\mathbf{n}$ the vector of all such counts. We assume for the moment the data set to be complete. The *likelihood* function can be expressed as a product of powers of the theta-parameters:

$$L(\theta|\mathbf{n}) \propto \prod_{c\in\Omega_C}\left[\theta_c^{n(c)}\prod_{i=1}^{k}\prod_{a_i\in\Omega_{A_i}}\theta_{a_i|c}^{n(a_i,c)}\right]. \tag{19}$$

Observe that for all $c\in\Omega_C$ and $i=1,\ldots,k$, the counts satisfy the *structural constraints* $0\le n(a_i,c)\le n(c)$, $\sum_{c\in\Omega_C}n(c)=n$ and $\sum_{a_i\in\Omega_{A_i}}n(a_i,c)=n(c)$, with n total number of instances.

The prior is usually expressed as a product of Dirichlet distributions. Under this choice, the prior is analogous to the likelihood, but the counts $n(\cdot)$ are replaced everywhere by $st(\cdot)-1$, where $s>0$ is the *equivalent sample size*, which can be interpreted as the number of hidden instances. The parameters $t(\cdot)$ can be interpreted as the proportion of units of the given type; for instance, $t(c')$ is the proportion of hidden instances for which $C=c'$, while $t(a_i,c')$ is the proportion of hidden instances for which $C=c'$ and $A=a_i$. This is a non-standard parameterization of the Dirichlet distribution, introduced in [77] because of its convenience when dealing with the IDM; the usual parameterization is instead $\alpha(\cdot)=st(\cdot)$.

We consider in particular the Perks prior, as in [81, Section 5.2]:

$$t(c) = \frac{1}{|\Omega_C|}; \; t(a_i,c) = \frac{1}{|\Omega_C||\Omega_{A_i}|}. \tag{20}$$

However, in some cases the uniform prior is modeled adopting the Laplace estimator [79, Chapter 4.2], which is different from Equation (20): it sets $\alpha(c)=st(c)=1\;\forall c$ and

XXVI

$\alpha(a_i, c) = st(a_i, c) = 1 \ \forall c, i$, which corresponds to initialize all counts $n(c)$ and $n(a_i, c)$ to 1 before analyzing the data. For instance, the WEKA implementation [79] of NBC is done in this way. However, there are slightly different versions also for the Laplace estimator; see for instance [56].

By multiplying the prior density and the likelihood function, we obtain a posterior density for $\theta_{c,\mathbf{a}}$, which is again a product of independent Dirichlet densities:

$$P(\theta_{c,\mathbf{a}}|\mathbf{n},\mathbf{t}) \propto \prod_{c \in \Omega_C} \left[\theta_c^{n(c)+st(c)-1} \prod_{i=1}^{k} \prod_{a_i \in \Omega_{A_i}} \theta_{a_i|c}^{n(a_i,c)+st(a_i,c)-1} \right]. \tag{21}$$

compared to the likelihood (19), the parameters $n(\cdot)$ are replaced by $n(\cdot) + st(\cdot)$. The joint probability of c and $\mathbf{a}$ can be computed by taking expectation from the posterior :

$$P(c,\mathbf{a}|\mathbf{n},\mathbf{t}) = E(c,\mathbf{a}|\mathbf{n},\mathbf{t}) = P(c|\mathbf{n},\mathbf{t}) \prod_{i=1}^{k} P(a_i|c,\mathbf{n},\mathbf{t}) \tag{22}$$

where

$$P(c|\mathbf{n},\mathbf{t}) = E[\theta_c|\mathbf{n},\mathbf{t}] = \tfrac{n(c)+st(c)}{n+s}, \tag{23}$$

$$P(a_i|c,\mathbf{n},\mathbf{t}) = E[\theta_{a_i|c}|\mathbf{n},\mathbf{t}] = \tfrac{n(a_i,c)+st(a_i,c)}{n(c)+st(c)}. \tag{24}$$

A problem of NBC, and more in general of any Bayesian classifier, is that sometimes the classifications is *prior-dependent*, namely the most probable class varies with the parameters $t(\cdot)$. Most often, one chooses the uniform prior trying to be non-informative; yet, we have already argued that prior-dependent classifications are fragile and that the uniform prior does not satisfactorily model prior ignorance. To address this issue, NCC is based on a set of priors rather than on a single prior.

9 Naive Credal Classifier (NCC)

NCC extends NBC to imprecise probabilities by considering a (credal) set of prior densities, instead of a unique prior. This prior credal set is modeled through the Imprecise Dirichlet Model (IDM) [77] and expresses prior *near-ignorance* [77, Section 4.6.9]; [17] it is then turned into a set of posteriors (posterior credal set) by element-wise application of Bayes' rule.

NCC specifies a *joint* credal set using the IDM; this is obtained by allowing each parameter of type $t(\cdot)$ to range within an interval, rather than being fixed to a single value. In particular the IDM contains all the densities for which $\mathbf{t}$ varies within the polytope T, defined as follows:

$$T = \begin{cases} \sum_{c \in \Omega_C} t(c) = 1 \\ t(c) > 0 & \forall c \in \Omega_C \\ \sum_{a \in \Omega_A} t(a,c) = t(c) & \forall c \in \Omega_C \\ t(a,c) > 0 & \forall a \in \Omega_A, c \in \Omega_C. \end{cases} \tag{25}$$

---

[17] Indeed, full ignorance is not compatible with learning; see Section 7.3.7 of [77] and [86].

Such constraints are analogous to the structural constraints which characterize the counts $n(\cdot)$. The third constraint introduces a link between the credal set $K(C)$ and the credal sets $K(A_i|c)$, with $c \in \Omega_C$, so that the corresponding credal network is *not* separately specified. Since the $t(\cdot)$ vary within an interval, also the posterior probability of class c lies within an interval. For instance, the upper and lower probability of c and $\mathbf{a}$ are:[18]

$$\underline{P}(c,\mathbf{a}|\mathbf{n},\mathbf{t}) := \inf_{\mathbf{t} \in T} P(c,\mathbf{a}|\mathbf{n},\mathbf{t})$$

$$\overline{P}(c,\mathbf{a}|\mathbf{n},\mathbf{t}) := \sup_{\mathbf{t} \in T} P(c,\mathbf{a}|\mathbf{n},\mathbf{t}).$$

While a traditional classifier returns the class with the highest posterior probability, credal classifiers return the classes which are *non-dominated*. The two criteria introduced in Section 5.1 can be considered to assess whether class c' dominates class c''. According to *interval-dominance* c' dominates c'' if $\underline{P}(c',\mathbf{a}|\mathbf{n},\mathbf{t}) > \overline{P}(c'',\mathbf{a}|\mathbf{n},\mathbf{t})$. Instead, according to *maximality*, c' dominates c'' if $P(c',\mathbf{a}|\mathbf{n},\mathbf{t}) > P(c'',\mathbf{a}|\mathbf{n},\mathbf{t})$ for all the values of $\mathbf{t} \in T$. Maximality is more powerful than interval-dominance, because it sometimes detect dominances which cannot be spotted using interval-dominance. Once the dominance criterion is chosen, the set of non-dominated classes is identified through repeated pairwise comparisons, as shown in the following pseudo-code:

```
// Since NCC is based on maximality, we denote the set
// of non−dominated classes as Ω_C^**.
// With interval−dominance, it should be denoted as Ω_C^*.

Ω_C^**  := Ω_C;
for   c' ∈ Ω_C{
   for   c'' ∈ Ω_C,  c'' ≠ c'{
      if (c' dominates c''){
         remove c'' from Ω_C^**;
      }
   }
}
return Ω_C^**;
```

In the following we sketch the test of dominance for NCC under maximality, designed in [81]. According to maximality, c' dominates c'' iff:

$$\inf_{\mathbf{t} \in T} \frac{P(c',\mathbf{a}|\mathbf{n},\mathbf{t})}{P(c'',\mathbf{a}|\mathbf{n},\mathbf{t})} > 1, \tag{26}$$

---

[18] Unlike Equation (3), these optimizations are over the open polytope T. For this reason, infima and suprema are considered instead of minima and maxima.

and assuming $P(c_2, \mathbf{a}|\mathbf{n}, \mathbf{t}) > 0$. Problem (26) can be re-written [81] considering Equations (23–24) as:

$$\inf_{\mathbf{t} \in T} \left\{ \left[\frac{n(c'') + st(c'')}{n(c') + st(c')} \right]^{k-1} \prod_{i=1}^{k} \frac{n(a_i, c') + st(a_i, c')}{n(a_i, c'') + st(a_i, c'')} \right\}. \tag{27}$$

As proved in [81], the infimum of problem (27) is obtained by letting $t(a_i, c') \to 0$ and $t(a_i, c'') \to t(c'')$. The values of these parameters at the optimum are extreme, as they touch the boundary of the IDM. The remaining parameters $t(c')$ and $t(c'')$ are optimized by noting that the infimum is achieved when $t(c') + t(c'') = 1$, which allows to express $t(c'')$ as $1 - t(c'')$. The final step to solve problem (27) involves a convex optimization over the single parameter $t(c')$; see [81] for more details.

The classification is *determinate* or *indeterminate* if there are respectively one or more non-dominated classes. The set of non-dominated classes returned by NCC always contains the most probable class identified by NBC, as the uniform prior is included in the IDM; [19] this also means that NCC, when determinate, returns the same class of NBC. On the other hand, if there are more non-dominated classes, the classification issued by NBC is prior-dependent. The non-dominated classes are *incomparable* and thus cannot be further ranked.

NCC has been originally introduced by [81]; applications of NCC to real-world case studies include diagnosis of dementia [87] and prediction of presence of parasites in crops [83]; it has been then extended with a sophisticated treatment of missing data in [29].

In the following, we show a comparison of NBC and NCC in texture recognition; the data set is complete and thus indeterminate classifications are only due to prior-dependent instances.

9.1 Comparing NBC and NCC in Texture Recognition

The goal of texture classification is to assign an unknown image to the correct texture class; this requires an effective description of the image (i.e., obtaining good features) and a reliable classifier. Texture classification is used in many fields, among which industrial applications, remote sensing and biomedical engineering. We compare NBC and NCC on the public OUTEX [62] data set of textures; the results presented in this section are taken from [26], where more details and experiments are described. The data set contains 4500 images from 24 classes of textures, including different kinds of canvas, carpets, woods etc.; some samples are shown in Fig. 12. We use the standard Local Binary Patterns (LBP) [62] as descriptors; they are computed by assigning each pixel to a category comprised between 1 and 18, on the basis of a comparison between its gray level and the gray level of the neighboring pixels. The features of an image are constituted by the percentage of pixels assigned to the different categories; therefore, 18 features are created for each image. There are no missing data.

---

[19] This is guaranteed with the Perks prior of Equation (20), but not with the Laplace estimator, which is not included into the IDM; yet, empirically this is most often the case also with the Laplace estimator.
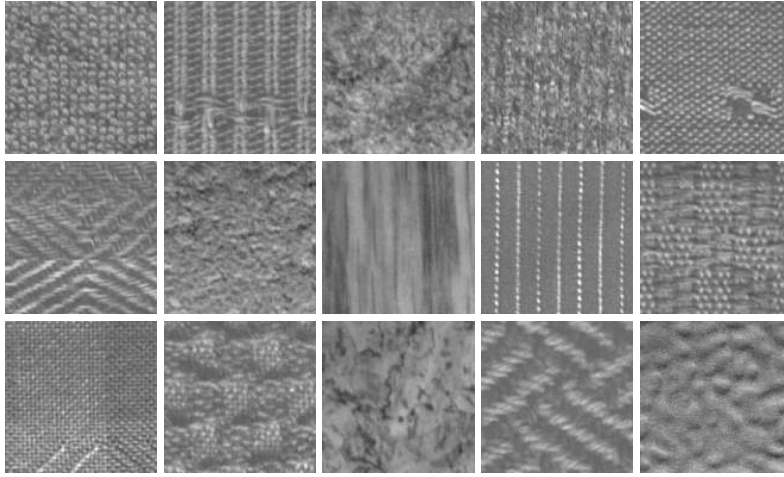
Fig. 12. Examples of some textures: each image refers to a different class.

We evaluate the classifiers through cross-validation, discretizing the features via supervised discretization [42]; the feature are discretized on average in some 10 bins. As our aim is to compare NBC and NCC rather than finely tuning the classifiers for maximum performance, we do not perform feature selection.

NBC achieves 92% accuracy, which can be considered satisfactory: for instance, SVMs are only slightly better, achieving 92.5%. However, NBC is unreliable on the prior-dependent instances, which amount to about 5% of the total. On these instances, NBC achieves only 56% accuracy, while NCC returns on average 2.5 classes, achieving 85% accuracy. On the non-prior dependent instances, both NBC and NCC achieve 94% accuracy.

Prior-dependent instances are present even on this large data set because each conditional probability distribution of type $P(A_j|C)$ requires to estimate some 240 parameters (24 classes * 10 states of the feature after discretization); since some combinations of the value of the class and of the feature rarely appear in the data set, the estimate of their probability is sensitive on the chosen prior.

Experiments at varying size of the training set are presented in Fig.13. At each size of the training set, NBC is characterized by a mix of good accuracy on the instances which are not prior-dependent, and bad accuracy on the prior-dependent ones; see Fig.13(a). Thanks to indeterminate classifications, NCC is instead much more accurate than NBC on the prior-dependent instances: see Fig.13(b). With increasing size of the training set, NCC becomes more determinate, steadily reducing both the percentage of indeterminate classification and the average number of classes returned when indeterminate; see Fig. 13(c).

Summing up, NBC is generally little accurate on the instances indeterminately classified by NCC; NCC preserves its reliability on prior-dependent instances thanks to indeterminate classifications; the determinacy of NCC increases with the size of the training set; indeterminate classifications can convey valuable information, when a small
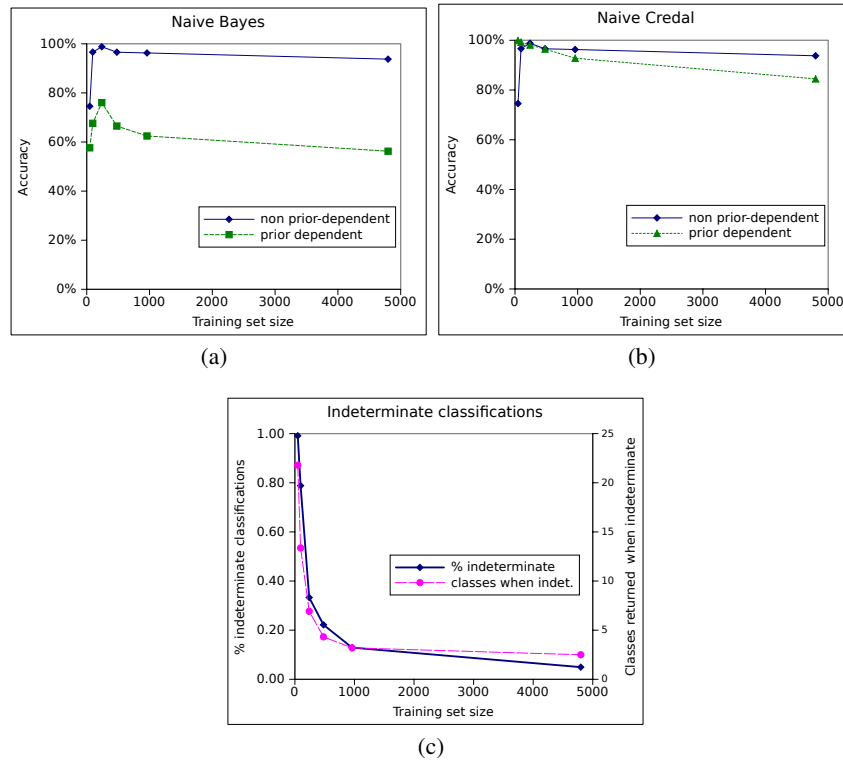
Fig. 13. Experiments with varying sizes of the training set. Plots (a) and (b) show the accuracy of NBC and NCC on instances which are prior dependent (dashed) and non prior-dependent (solid); plot (c) shows the percentage of indeterminate classifications (solid) and the average number of classes returned by NCC when indeterminate (dashed).

subset of classes is returned out of many possible ones. Such results are consistent with those obtained [29] on the classical UCI data sets.

However, NBC provides no way of understanding whether a certain instance is prior-dependent. One could try to mimic the behavior of NCC by setting a *rejection rule* on NBC, outputting more classes if the probability of the most probable class does not exceed a certain threshold. Yet, a rejection rule is likely to be little effective with NBC, which generally returns high probability for the most probable class. In the texture application, NCC detects about half of the prior-dependent instances among those which are classified by NBC with probability higher than 95%. As discussed in [29, Section 4.4], an instance is less likely to be prior-dependent as the probability computed by NBC for the most probable class increases, but such correlation is not deterministic: there are prior dependent instances classified by NBC with high probability, and non-prior dependent ones classified by NBC with a relatively small margin. Overall, the prior-dependency analysis performed by NCC is much more sophisticated than any rejection rule.

9.2 Treatment of Missing Data

Very often, real data sets are *incomplete*, because some values of the feature variables are not present.[20] Dealing with incomplete data sets rests on the assumptions done about the process responsible for the missingness. This process can be regarded as one that takes in input a set of complete data, which is generally not accessible for learning, and that outputs an incomplete data set, obtained by turning some values into missing. Learning about the missingness process' behavior is usually not possible by only using the incomplete data. This fundamental limitation explains why the assumptions about the missingness process play such an important role in affecting classifiers' predictions. Moreover, a missingness process may also be such that the empirical analysis of classifiers is doomed to provide misleading evidence about their actual predictive performance, and hence, indirectly, about the quality of the assumptions done about the missingness process. This point in particular has been discussed in [29, Section 4.6] and [86, Section 5.3.2]. For these reasons, assumptions about the missingness process should be stated with some care.

In the vast majority of cases, common classifiers deal with missing values (sometimes implicitly) assuming that the values are MAR [69]. However, assuming MAR when it does not hold can decreases the classification accuracy.

The NCC has been one of the first classifiers [81, Section 3], together with Ramoni and Sebastiani's *robust Bayes classifier* [65] (a robust variant of NBC to deal with missing data), to provide a way to conservatively deal with non-MAR missing data in the training set. Both approaches are based on very weak, and hence tenable, assumptions about the missingness process; in fact, they regard as possible any realization of the training set, which is consistent with the incomplete training set; this way of dealing with missing data has been pioneered in statistics by Manski [60].

In particular, according to the conservative inference rule, introduced in Section 5.3, conservative treatment of missing data requires to compute many likelihoods, one per each complete data sets consistent with the incomplete training set. In particular, the approach of [81] is equivalent to inferring many NCCs: one per each complete data sets consistent with the incomplete training set; the classification is given by the union of the set of non-dominated classes produced by all the NCCs.[21] In [81] specific procedures are designed, which perform the computation exactly and in linear time w.r.t. the amount of missing data, avoiding to enumerate the (exponentially many) complete data sets and to infer many NCCs. The imprecision introduced by missing data leads to an increase in the indeterminacy of the NCC, which is related to the amount of missingness. In other words, the NCC copes with the weak knowledge about the missingness process by weakening the answers to maintain reliability.

In [29] the treatment of missing data has further improved, allowing NCC to deal with non-MAR missing data also in the instance to classify and not only in the training set, and to deal with a mix of MAR and non-MAR features (treating the first group according to MAR and the second in a conservative way), using CIR. The resulting classifier is called NCC2. Distinguishing variables that are subject to the two types of

---

[20] We do not consider here the case of missing values of the class variable.

[21] Strictly speaking, this straightforward explanation is valid for the case of two classes.

processes is important because treating MAR variables in a conservative way leads to an excess of indeterminacy in the output that is not justified. In fact, the experimental results of NCC2 [29] show that the indeterminacy originated from missing data is compatible with informative conclusions provided that the variables treated in a conservative way are kept to a reasonable number (feature selection can help in this respect, too). Moreover, they show that the classifiers that assume MAR for all the variables are often substantially unreliable when NCC2 is indeterminate.

Formal justifications of the rule NCC2 uses to deal with missing values can be found in [29]. This work discusses also, more generally, the problem of incompleteness for uncertain reasoning.

10   Metrics for Credal Classifiers

Before introducing further credal classifiers, it is useful to review the metrics which can be used to compare them. The overall performance of a credal classifier can be fully characterized by four indicators [29]:

- *determinacy*, i.e., the percentage of instances determinately classified;
- *single-accuracy*, i.e., the accuracy on the *determinately* classified instances;
- *set-accuracy*, i.e., the accuracy on the *indeterminately* classified instances;
- *indeterminate output size*: the average number of classes returned on the indeterminately classified instances.

However, set-accuracy and indeterminate output size are meaningful only if the data set has more than two classes.

These metrics completely characterize the performance of a credal classifier, but do not allow to readily compare two credal classifiers. Two metrics suitable to compare credal classifiers have been designed in [30]. The first one, borrowed from multi-label classification,[22] is the *discounted-accuracy*:

$$\text{d-acc} = \frac{1}{n_{te}} \sum_{i=1}^{n_{te}} \frac{(accurate)_i}{|Z_i|},$$

where $(accurate)_i$ is a 0-1 variable, showing whether the classifier is accurate or not on the i-th instance; $|Z_i|$ is the number of classes returned on the i-th instance and n_{te} is the number of instances of the test set. However, discounting *linearly* the accuracy on the output size is arbitrary. For example, one could instead discount on $|Z_i|^2$.

The non-parametric *rank test* overcomes this problem. On each instance, it ranks two classifiers CL_1 and CL_2 as follows:

- if CL_1 is accurate and CL_2 inaccurate: CL_1 wins;
- if both classifiers are accurate but CL_1 returns less classes: CL_1 wins;
- if both classifiers are wrong: tie;
- if both classifiers are accurate with the same output size: tie.

---

[22] The metric is referred to as *precision* in [73].

The wins, ties and losses are mapped into ranks and then analyzed via the Friedman test. The rank test is more robust than d-acc, as it does not encode an arbitrary function for the discounting; yet, it uses less pieces of information and can therefore be less sensitive. Overall, a cross-check of the both indicators is recommended.

Instead, an open problem is how to compare a credal classifier with a classifier based on traditional probability. So far, this comparison has been addressed by comparing the accuracy achieved by the Bayesian classifier on the instances determinately and indeterminately classified by the credal classifier, thus assessing how good the credal classifier is at isolating instances which cannot be safely classified with a single class. This produces a statistics of type: on the prior-dependent instances, the Bayesian classifier achieves 60% accuracy returning a single class, while the credal classifier achieves 90% accuracy, returning two classes. But which one is better? Moreover, returning a very similar probability for the most probable and the second most probable class (for the Bayesian) should be considered equivalent to returning two classes (for the credal). A metric able to rigorously compare credal and Bayesian classifier could be very important to allow credal classifiers to become widespread.

11 Tree-Augmented Naive Bayes (TAN)

In [47], NBC and Bayesian Networks (BNs) whose topology has been learned from data, have been compared in classification; surprisingly BNs, despite their much higher flexibility, did not outperform NBC. This can be explained through the bias-variance decomposition of the misclassification error, which we already mentioned in Section 8: BNs have much lower bias than NBC, but this effect is often not felt, because of their high variance. However, these results were the inspiration for developing an effective compromise between BNs and NBC, yielding the so-called *tree-augmented naive Bayes* (TAN) [47], which is defined as follows (see also Figure 14):

- each feature has the class as a parent;
- each feature can also have an additional second parent, constituted by another feature.
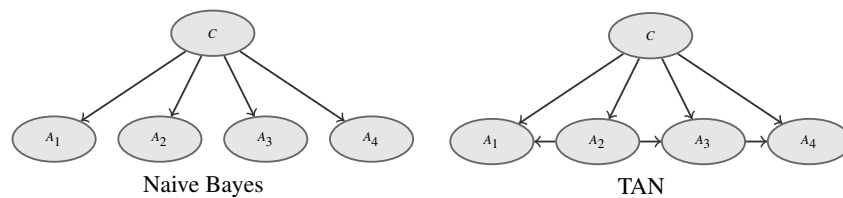

Fig. 14. TAN can model dependencies between features, unlike naive Bayes.

In [47], TAN has been shown to be generally more accurate than both NBC and BNs. More recent results [59] point out a flaw regarding the usage of BNs in [47]; however, even after fixing this problem, the results confirm that TAN is generally more

accurate than both NBC and BNs (although the advantage of TAN over BNs is less marked than previously reported, and moreover BNs are now shown to be indeed more accurate than NBC).

This justifies the interest for designing a credal TAN. Before reviewing its development it is however necessary to discuss the different variants of the IDM which can be used for classification.

11.1   Variants of the Imprecise Dirichlet Model: Local and Global IDM

Given a credal network, three kinds of IDM can be used: the *global*, the *local* and the recently introduced Extreme Dirichlet Model (EDM) [16]. In the following, we show the differences between these approaches, using the example network $C \rightarrow A$.

Let us focus on the class node. The constraints which define the set of Dirichlet distributions for the IDM (both local and global) are:

$$T_C = \begin{cases} \sum_{c \in \Omega_C} t(c) = 1 \\ t(c) > 0 & \forall c \in \Omega_C. \end{cases} \tag{28}$$

As in Equation (7), the credal set $K(C)$ contains the mass functions of type $P(C)$, which allows the probability of class c to vary within the interval:

$$P(c) \in \left[\frac{n(c)}{s + \sum_{c \in \Omega_C} n(c)}, \frac{s + n(c)}{s + \sum_{c \in \Omega_C} n(c)} \right]. \tag{29}$$

The EDM restricts the set of priors defined by Eq.(28) to its most extreme elements, i.e., each $t(c)$ can be only zero or one. Consequently, the probability of class c corresponds either to the upper or to the lower bound of the interval in Equation (29).

Let us now move to conditional probabilities. The local IDM defines the polytope similarly to Equation (28):

$$T_{A|C} = \begin{cases} \sum_{a \in \Omega_A} t(a,c) = 1 & \forall c \in \Omega_C \\ t(a,c) > 0 & \forall a \in \Omega_A, \forall c \in \Omega_C. \end{cases} \tag{30}$$

Note that there is no relation between the $t(a,c)$ and the $t(c)$ previously used for the class node. For each $c \in \Omega_C$, the credal set $K(A|c)$ contains the mass functions of type $P(A|c)$, which let its probabilities vary as follows:

$$P(a|c) \in \left[\frac{n(a,c)}{s + n(c)}, \frac{s + n(a,c)}{s + n(c)} \right]. \tag{31}$$

The credal sets $\{K(A|c)\}_{c \in \Omega_C}$ and $K(C)$ are thus specified one independently of the others. Following the terminology of Section 4.2, the model is a *separately specified* credal network.

Instead, to understand the estimate of the conditional probabilities under the *global* IDM, we should recall that it is based on a set of *joint* Dirichlet distributions, defined

by the constraints (already given in Section 8):

$$T_{A,C} = \begin{cases} \sum_{c \in \Omega_C} t(c) = 1 \\ t(c) > 0 & \forall c \in \Omega_C \\ \sum_a t(a,c) = t(c) & \forall c \in \Omega_C \\ t(a,c) > 0 & \forall a \in \Omega_A, \forall c \in \Omega_C. \end{cases} \qquad (32)$$

In particular, the third constraint introduces a link between $t(a,c)$ and $t(c)$, which is missing in the local IDM; therefore, the network is *not* separately specified. Given the value of $t(c)$, the credal set $K(A|c)$ contains the mass functions $P(A|c)$ such that:

$$P(a|c) \in \left[\frac{n(c,a)}{st(c) + n(c)}, \frac{st(c) + n(c,a)}{st(c) + n(c)} \right]. \qquad (33)$$

The global IDM estimates narrower intervals than the local, as can be seen by comparing Equation (33) and Equation (31)[23]: this implies less indeterminacy in classification. Yet, the global IDM poses challenging computational problems; so far, exact computation with the global IDM has been possible only with NCC. Instead, the local IDM can be computed for any network and is in fact the common choice for general credal networks; yet, it returns wider intervals.

The EDM restricts the *global* IDM to its extreme distributions; it therefore allows $t(a,c)$ to be either 0 or $t(c)$, keeping the constraint $\forall c \in \Omega_C : \sum_{a \in \Omega_A} t(a,c) = t(c)$ inherited from the global IDM. The extreme points of the EDM corresponds in this case to the bounds of the interval in Equation (33); but in general, they are a inner approximation of the extremes of the global IDM [16]. From a different viewpoint, the EDM can be interpreted as treating the *s* hidden instances as *s* rows of non-MAR missing data, but with the additional assumption that such rows are all *identical* to each other; ignorance is due to the fact that it is unknown which values they contain.

The approximation provided by the EDM has been experimentally validated [25] by comparing the classification produced by NCC under the global IDM and the EDM; NCC produces almost identical results in the two settings, and thus the EDM appears as a reliable approximation of the global IDM, with the advantage of a simplified computation.

12   Credal TAN

As already discussed, the computational problems posed by the global IDM are quite challenging and imply a large computational overload for non-naive topologies. Thus, over years alternative solutions have been investigated.

A credal TAN was firstly proposed in [84], using the local IDM. The classifier was indeed reliable and very accurate when returning a single class but it was excessively cautious because of the local IDM. We refer to this algorithm as TANC*.

In [25], a credal TAN has been designed using the EDM; we refer this algorithm as TANC. As shown in Fig.15(a), TANC is more determinate than TANC*, because the

---

[23] Recall that $\sum_c t(c) = 1$ and that $t(c) > 0 \; \forall c \in \Omega_C$.

EDM is an inner approximation of the global IDM, which in turn computes narrower intervals than the local IDM. More important, TANC consistently achieves higher discounted accuracy than TANC*, as shown in Fig.15(b); therefore, it realizes a better trade-off between informativeness and reliability. However, the two classifiers have the same performance on the kr-kp data set, which contains few thousands of instances and only binary features; in this case, the model of prior ignorance has little importance.
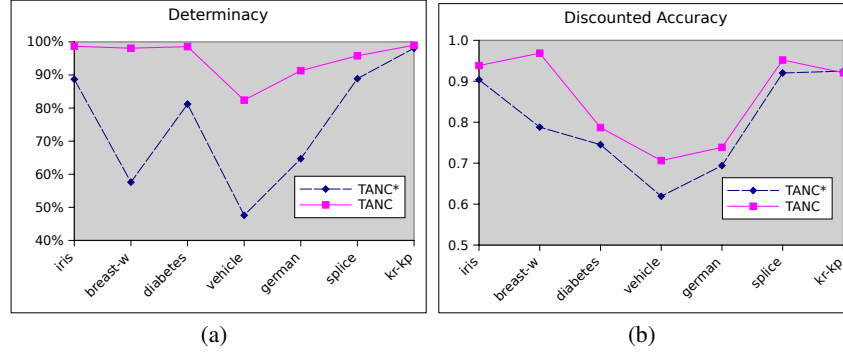

(a)   (b)

Fig. 15. Comparison of TANC* and TANC. Plot (a) shows the determinacy (% of determinate classifications) of the classifiers on different data sets, while plot (b) shows their discounted accuracy.

TANC is moreover good at spotting instances over which the Bayesian TAN becomes unreliable, similarly to how NCC does with NBC. In [25], experiments over some 40 UCI data sets show an average drop of 30 points of accuracy for the Bayesian TAN between the instances determinately and indeterminately classified by TANC. Instead, TANC preserves reliability also on the prior-dependent instances, [24] thanks to indeterminate classifications.

Although TANC is consistently more determinate than TANC*, it becomes sometimes largely indeterminate, especially on small data sets characterized by many classes and/or categorical values of the features. In fact, the TAN architecture (learned using a MDL criterion implemented within WEKA [79]), sometimes assigns the second parent to a feature, even though the resulting contingency table contains many counts which are numerically small. When parsed by TANC, they generate prior-dependent classifications and thus indeterminacy.

This also causes TANC to be slightly outperformed by NCC, as shown by the scatter-plot of the discounted accuracy of Fig.16; therefore, TANC looses the advantage which the Bayesian TAN has over NBC. In [25], it is hypothesized that an algorithm for learning the structure more suitable for TANC should return less parameterized structures and could allow a significant performance improvement. Such algorithm should

---

[24] Note that different credal classifiers, encoding a different probabilistic assumptions, might judge the same instance as prior-dependent or not. Thus, an instance is *not* prior-dependent per se, but according to the judgment of a certain credal classifier.

be able to return even a naive structure, if for instance modelling further dependencies makes the joint distribution too sensitive on the prior. Previous attempts for structure learning based on imprecise probability can be found in [85]; yet this field has not been extensively explored and constitutes an interesting area for future research.
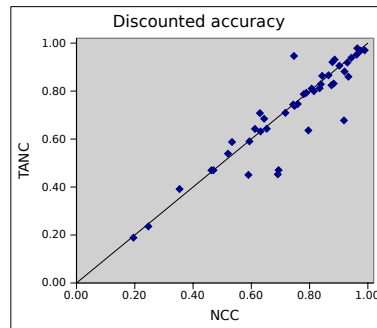

Fig. 16. Discounted accuracy of TANC and NCC.

TANC is moreover able to conservatively deal with non-MAR missing data [25]. However, the treatment of missing data is at an earlier stage compared to that of NCC, as all missing data are currently treated as non-MAR (it is currently no possible to deal with a mix of MAR and non-MAR features) and moreover the current algorithms are not yet developed to deal with non-MAR missing data in the instance to classify. Preliminary results [25] show that, when faced with incomplete training sets, TANC is much more indeterminate than NCC but achieves a similar discounted accuracy.

13   Further Credal Classifiers

13.1   Lazy NCC (LNCC)

Besides the TAN approach, a further possibility of reducing the bias due to the naive assumption is to combine NCC and *lazy learning*; this has been explored in [30].

Lazy learning defers the training, until it has to classify an instance (*query*). In order to classify an instance, a lazy algorithm:

1. ranks the instances of the training set according to the distance from the query;
2. trains a local classifier on the k instances nearest to the query and returns the classification using the local classifier;
3. discards the locally trained classifier and keeps the training set in memory in order to answer new queries.

Lazy classifiers are *local*, as they get trained on the subset of instances which are nearest to the query. The parameter k (*bandwidth*) controls the bias-variance trade-off for lazy learning. In particular, a smaller bandwidth implies a smaller bias (even a simple model can fit a complex function on a small subset of data) at a cost of a larger

variance (as there are less data for estimating the parameters). Therefore, learning locally NBC (or NCC) can be a winning strategy as it allows reducing the bias; moreover, it also reduces the chance of encountering strong dependencies between features [45]. In fact, a successful example of lazy NBC is given in [45].

However, an important problem dealing with lazy learning is how to select the bandwidth k. The simplest approach is to empirically choose k (for instance, by cross-validation on the training set) and to then use the same k to answer all queries. However, the performance of lazy learning can significantly improve if the bandwidth is adapted query-by-query, as shown in [12] in the case of regression.

LNCC tunes the bandwidth query-by-query using a criterion based on imprecise probability. After having ranked the instances according to their distance from the query, a local NCC is induced on the k_{min} closest instances (for instance, $k_{min} = 25$) and classifies the instance. The classification is accepted if determinate; otherwise, the local NCC is updated by adding a set of further k_{upd} instances (we set $k_{upd} = 20$) to its training set. The procedure continues until either the classification is determinate or all instances have been added to the training of the local NCC. Therefore, the bandwidth is increased until the locally collected data smooth the effect of the choice of the prior. The naive architecture makes it especially easy updating LNCC with the k_{upd} instances; it only requires to update the counts $n(\cdot)$ that are internally stored by LNCC.

By design LNCC is thus generally more determinate than NCC; this also helps addressing the excessive determinacy which sometimes characterizes also NCC [24]. In [30] that generally LNCC outperforms NCC, both according to the discounted accuracy and the rank test.

13.2 Credal model averaging (CMA)

Model uncertainty is the problem of having multiple models which provide a good explanation of the data, but lead to different answers when used to make inference. In this case, selecting a single model underestimates uncertainty, as the uncertainty about model selection is ignored. *Bayesian model averaging* (BMA) [52] addressed model uncertainty by averaging over a set of candidate models rather than selecting a single candidate; each model is given a weight corresponding to its posterior probability.

In case of NBC, given k features, there are 2^k possible NBCs, each characterized by a different subset of features; we denote by $\mathcal{M}$ the set of such models and by m a generic model of the set. Using BMA, the posterior probability $P(c, \mathbf{a}|\mathbf{n}, \mathbf{t})$ is computed by averaging over *all* the 2^k different NBCs, namely by marginalizing m out:

$$P(c, \mathbf{a}|\mathbf{n}, \mathbf{t}) \propto \sum_{m \in \mathcal{G}} P(c, \mathbf{a}|\mathbf{n}, \mathbf{t}, m) P(\mathbf{n}|m) P(m), \qquad (34)$$

where $P(m)$ and $P(\mathbf{n}|m) = \int P(\mathbf{n}|m, \mathbf{t}) P(\mathbf{t}|m) d\mathbf{t}$ are respectively the prior probability and the marginal likelihood of model m; the posterior probability of model m is $P(m|\mathbf{n}, \mathbf{t}) \propto P(\mathbf{n}|m, \mathbf{t}) P(m|\mathbf{t})$.

BMA implies two main challenges [19]: the computation of the exhaustive sum of Eq.(34) and the choice of the prior distribution over the models.

The computation of BMA is difficult, because the sum of Eq. (34) is often intractable; in fact, BMA is often computed via algorithms which are both approximated

and time-consuming. However, Dash and Cooper [36] provide an exact and efficient algorithm to compute BMA over 2^k NBCs.

As for the choice of the prior, a common choice is to assign equal probability to all models; however, this is criticized from different standpoints even in the literature of BMA (see the rejoinder of [52]). Moreover, as already discussed, the specification of any single prior implies arbitrariness and entails the risk of issuing prior-dependent classifications. Our view is that this problem should be addressed by using a credal set rather than a single prior. However, in the following it is understood that by BMA we mean BMA learned with the uniform prior over the models.

Credal set Credal model averaging (CMA) [27] extends to imprecise probabilities the BMA over NBCs of [36], substituting the single prior over the models by a credal set. The prior probability of model m is expressed by Dash and Cooper [36] as:

$$P(g) = \prod_{i \in m} P_i \prod_{i \notin m} (1 - P_i), \tag{35}$$

where P_i is the probability of feature i to be relevant for the problem, while $i \in g$ and $i \notin g$ index respectively the features included and excluded from model g. By setting $P_i := 0.5$ for all i, all models are given the same prior probability.

CMA is aimed at modelling a condition close to *prior ignorance* about the relative credibility of the 2^k NBC$_s$, which also implies ignorance about whether each feature is relevant or not; the credal set $K(M)$ of prior over the models is given by all the mass function obtained by letting vary each P_i within the interval $\varepsilon < P_i < 1 - \varepsilon$ (the introduction of the $\varepsilon > 0$ is necessary to enable learning from the data).

Denoting as $P(M)$ a generic mass function over the graphs, the test of credal-dominance test of CMA is:

$$\inf_{P(M) \in K(M)} \frac{\sum_{g \in \mathscr{G}} P(c_1|g,\mathbf{n})P(\mathbf{n}|g)P(g)}{\sum_{g \in \mathscr{G}} P(c_2|g,\mathbf{n})P(\mathbf{n}|g)P(g)} > 1. \tag{36}$$

The computation of the dominance test is accomplished by extending to imprecise probability the BMA algorithm by [36]; see [27] for more details.

Since $K(M)$ contains the uniform prior over the models, the set of non-dominated classes of CMA always contains the most probable class identified by BMA; for the same reasons CMA, when determinate, returns the same class of BMA.

The experiments of [27] shows that the accuracy of BMA sharply drops on the instances where CMA gets indeterminate. The finding that a Bayesian classifier is little accurate on the instances indeterminately classified by its counterpart based on imprecise probabilities is indeed consistent across the various credal classifiers which we have developed.

A possible research direction is the development of CMA for NCC, namely imprecise averaging over credal classifiers. Yet, attempts in this direction seem to involve quite difficult and time-consuming computations.

14   Open Source Software

JNCC2 [28] is the Java implementation of NCC; it is available from `www.idsia.ch/`
`~giorgio/jncc2.html` and has a command-line interface. This software has been
around since some years and is stable.

A second open-source software is a plug-in for the WEKA [79] environment; it im-
plements NCC, LNCC, CMA and the credal version of classification trees [1]. Thanks
to the WEKA environment, all the operations with credal classifiers can be performed
graphically and moreover many powerful tools (e.g., feature selection) become avail-
able to be readily used with credal classifiers. This software is available from `http:`
`//decsai.ugr.es/~andrew/weka-ip.html`; it is very recent and thus should be
seen as more experimental.

Acknowledgements

15   Conclusions

Credal networks generalize Bayesian networks, providing a more robust probabilistic
representation; in some cases, a single probability distribution cannot robustly describe
uncertainty. Being able to work with a set of distributions rather than with a single dis-
tribution, credal networks can for instance robustly deal with the specification of the
prior and with non-MAR missing data. Credal networks are naturally suited to model
expert knowledge, as often the experts feel more confident in assigning to an event an
interval of probability rather than a point-wise probability; in fact, knowledge-based
systems are a natural application of credal networks. However, credal networks have
been also thoroughly developed for classification. The main feature of credal classi-
fiers is that they suspend the judgment returning a set classes; this happens for instance
when the instance is prior-dependent or when too much uncertainty arises from miss-
ing data, when MAR cannot be assumed. Extensive experiments, performed both on
public benchmark data sets and in real-world applications show that on the instances
indeterminately classified by a credal network, the accuracy of its Bayesian counterpart
(namely, a BN with the same graph, learned with the uniform distribution) drops. Direc-
tions for future research include the development of a more rigorous metric to compare
credal and traditional probabilistic classifier and algorithms for structure learning espe-
cially tailored for credal networks.

References

1. Abellán, J., Moral, S.: Upper entropy of credal sets. Applications to credal classification.
   International Journal of Approximate Reasoning 39(2-3), 235–255 (2005)
2. Antonucci, A., Brühlmann, R., Piatti, A., Zaffalon, M.: Credal networks for military identi-
   fication problems. International Journal of Approximate Reasoning 50(4), 666–679 (2009)

3. Antonucci, A., Cuzzolin, F.: Credal sets approximation by lower probabilities: Application to credal networks. In: IPMU 2010: Proceedings of the 13th Information Processing and Management of Uncertainty in Knowledge-Based Systems Conference (2010)

4. Antonucci, A., Piatti, A., Zaffalon, M.: Credal networks for operational risk measurement and management. In: Apolloni, B., Howlett, R.J., Jain, L.C. (eds.) KES (2). Lecture Notes in Computer Science, vol. 4693, pp. 604–611. Springer (2007)

5. Antonucci, A., Salvetti, A., Zaffalon, M.: 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 (2007)

6. Antonucci, A., Sun, Y., de Campos, C., Zaffalon, M.: Generalized loopy 2U: a new algorithm for approximate inference in credal networks. International Journal of Approximate Reasoning 51(5), 474–484 (2010)

7. Antonucci, A., Zaffalon, M.: 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.) Proceedings of third international conference on Soft Methods in Probability and Statistics (SMPS-2006). pp. 223–230. Springer (2006)

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

9. Avis, D., Fukuda, K.: Reverse search for enumeration. Discrete Applied Mathematics 65, 21–46 (1996)

10. Benavoli, A., de Campos, C.P.: Inference from multinomial data based on a MLE-dominance criterion. In: Proc. on European Conf. on Symbolic and Quantitative Approaches to Reasoning and Uncertainty (Ecsqaru). pp. 22–33. Verona (2009)

11. Benavoli, A., Zaffalon, M., Miranda, E.: Reliable hidden Markov model filtering through coherent lower previsions. In: Proc. 12th Int. Conf. Information Fusion. pp. 1743–1750. Seattle (USA) (2009)

12. Bontempi, G., Birattari, M., Bersini, H.: Lazy learning for local modelling and control design. International Journal of Control 72(7), 643–658 (1999)

13. de Campos, C.P., Cozman, F.G.: Inference in credal networks through integer programming. In: Proceedings of the Fifth International Symposium on Imprecise Probability: Theories and Applications. Action M Agency, Prague (2007)

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

15. Cano, A., Cano, J., Moral, S.: 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). pp. 4–8 (1994)

16. Cano, A., Gómez-Olmedo, M., Moral, S.: Credal nets with probabilities estimated with an extreme imprecise Dirichlet model. In: de Cooman, G., Vejnarová, I., Zaffalon, M. (eds.) Proceedings of the Fifth International Symposium on Imprecise Probability: Theories and Applications (ISIPTA '07). pp. 57–66. Action M Agency, Prague (2007)

17. Cano, A., Moral, S.: A review of propagation algorithms for imprecise probabilities. In: [38]. pp. 51–60 (1999)

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

19. Clyde, M., George, E.: Model uncertainty. Statistical Science pp. 81–94 (2004)

20. Coolen, F.P.A., Augustin, T.: Learning from multinomial data: a nonparametric predictive alternative to the Imprecise Dirichlet Model. In: ISIPTA'05: Proceedings of the Fourth International Symposium on Imprecise Probabilities and their Applications. pp. 125–134 (2005)

21. de Cooman, G., Hermans, F., Antonucci, A., Zaffalon, M.: Epistemic irrelevance in credal networks: the case of imprecise markov trees. International Journal of Approximate Reasoning (accepted for publication)
22. de Cooman, G., Miranda, E., Zaffalon, M.: Independent natural extension. In: IPMU 2010: Proceedings of the 13th Information Processing and Management of Uncertainty in Knowledge-Based Systems Conference (2010)
23. Cooper, G.F.: The computational complexity of probabilistic inference using Bayesian belief networks. Artificial Intelligence 42, 393–405 (1990)
24. Corani, G., Benavoli, A.: Restricting the IDM for classification. In: Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2010) (2010)
25. Corani, G., de Campos, C.P.: A tree-augmented classifier based on Extreme Imprecise Dirichlet Model. International Journal of Approximate Reasoning (accepted for publication)
26. Corani, G., Giusti, A., Migliore, D.: Robust texture recognition using imprecise classification. Under review
27. Corani, G., Zaffalon, M.: Credal model averaging: An extension of bayesian model averaging to imprecise probabilities. In: Proc. of the 2008 European Conf. on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD 08). pp. 257–271. Springer-Verlag (2008)
28. Corani, G., Zaffalon, M.: JNCC2: The Java implementation of naive credal classifier 2. Journal of Machine Learning Research 9, 2695–2698 (2008)
29. Corani, G., Zaffalon, M.: Learning reliable classifiers from small or incomplete data sets: the naive credal classifier 2. Journal of Machine Learning Research 9, 581–621 (2008)
30. Corani, G., Zaffalon, M.: Lazy naive credal classifier. In: Proc. of the 1st ACM SIGKDD Workshop on Knowledge Discovery from Uncertain Data. pp. 30–37. ACM (2009)
31. Costa, J.E., Fleisher, P.J. (eds.): Physical geomorphology of debris flows, chap. 9, pp. 268–317. Springer-Verlag, Berlin (1984)
32. Couso, I., Moral, S., Walley, P.: Examples of independence for imprecise probabilities. In: ISIPTA. pp. 121–130 (1999)
33. del Coz, J., Dıez, J., Bahamonde, A.: Learning Nondeterministic Classifiers. Journal of Machine Learning Research 10, 2273–2293 (2009)
34. Cozman, F.G.: Robustness analysis of Bayesian networks with finitely generated convex-sets of distributions. Tech. Rep. CMU-RI-TR 96-41, Robotics Institute, Carnegie Mellon University (1996)
35. Cozman, F.G.: Credal networks. Artificial Intelligence 120, 199–233 (2000)
36. Dash, D., Cooper, G.: Model averaging for prediction with discrete Bayesian networks. Journal of Machine Learning Research 5, 1177–1203 (2004)
37. de Campos, C.P., Cozman, F.G.: The inferential complexity of Bayesian and credal networks. In: Proceedings of the International Joint Conference on Artificial Intelligence. pp. 1313–1318. Edinburgh (2005)
38. de Cooman, G., Cozman, F.G., Moral, S., Walley, P. (eds.): ISIPTA '99: Proceedings of the First International Symposium on Imprecise Probabilities and Their Applications. The Imprecise Probability Project, Universiteit Gent, Belgium (1999)
39. de Cooman, G., Zaffalon, M.: Updating beliefs with incomplete observations. Artificial Intelligence 159, 75–125 (2004)
40. Domingos, P., Pazzani, M.: On the optimality of the simple Bayesian classifier under zero-one loss. Machine Learning 29(2/3), 103–130 (1997)
41. Elkan, C.: Magical thinking in data mining: lessons from CoIL challenge 2000. In: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 426–431. ACM (2001)

42. Fayyad, U.M., Irani, K.B.: Multi-interval Discretization of Continuous-valued Attributes for Classification Learning. In: Proc. of the 13th International Joint Conference on Artificial Intelligence. pp. 1022–1027. Morgan Kaufmann, San Francisco, CA (1993)

43. de Finetti, B.: 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

44. Fleuret, F.: Fast binary feature selection with conditional mutual information. Journal of Machine Learning Research 5, 1531–1555 (2004)

45. Frank, E., Hall, M., Pfahringer, B.: Locally weighted naive Bayes. In: Proceedings of the Conference on Uncertainty in Artificial Intelligence. pp. 249–256 (2003)

46. Friedman, J.: On bias, variance, 0/1 - loss, and the curse-of-dimensionality. Data Mining and Knowledge Discovery 1, 55–77 (1997)

47. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. Machine learning 29(2), 131–163 (1997)

48. Griffiths, P.G., Webb, R.H., Melis, T.S.: Frequency and initiation of debris flows in grand canyon, arizona. Journal of Geophysical Research 109, 4002–4015 (2004)

49. Ha, V., Doan, A., Vu, V., Haddawy, P.: Geometric foundations for interval-based probabilities. Annals of Mathematics and Artificial Intelligence 24(1–4), 1–21 (1998)

50. Hand, D., Yu, K.: Idiot's Bayes-Not So Stupid After All? International Statistical Review 69(3), 385–398 (2001)

51. Hoare, Z.: Landscapes of naive Bayes classifiers. Pattern Analysis & Applications 11(1), 59–72 (2008)

52. Hoeting, J., Madigan, D., Raftery, A., Volinsky, C.: Bayesian model averaging: A tutorial. Statistical science 14(4), 382–401 (1999)

53. Ide, J.S., Cozman, F.G.: IPE and L2U: Approximate algorithms for credal networks. In: Proceedings of the Second Starting AI Researcher Symposium. pp. 118–127. IOS Press, Amsterdam (2004)

54. Jaeger, M.: Ignorability for categorical data. Annals of statistics pp. 1964–1981 (2005)

55. Kohavi, R.: Scaling up the accuracy of naive-Bayes classifiers: A decision-tree hybrid. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. pp. 202–207. AAAI Press (1996)

56. Kohavi, R., Becker, B., Sommerfield, D.: Improving simple Bayes. In: Proc. 9th European Conference on Machine Learning (ECML '97'). pp. 78–87 (1997)

57. Levi, I.: The Enterprise of Knowledge. MIT Press, London (1980)

58. Little, R.J.A., Rubin, D.B.: Statistical Analysis with Missing Data. Wiley, New York (1987)

59. Madden, M.: On the classification performance of TAN and general Bayesian networks. Knowledge-Based Systems 22(7), 489–495 (2009)

60. Manski, C.F.: Partial Identification of Probability Distributions. Springer-Verlag, New York (2003)

61. Murphy, K., Weiss, Y., Jordan, M.: Loopy belief propagation for te inference: An empirical study. In: Conference on Uncertainty in Artificial Intelligence. pp. 467–475. Morgan Kaufmann, San Francisco (1999)

62. Ojala, T., Pietikainen, M., Maenpaa, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence pp. 701–706 (2002)

63. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, San Mateo, California (1988)

64. Piatti, A., Antonucci, A., Zaffalon, M.: Building knowledge-based systems by credal networks: a tutorial. In: Baswell, A.R. (ed.) Advances in Mathematics Research, vol. 11. Nova Science Publishers, New York (2010)

65. Ramoni, M., Sebastiani, P.: Robust learning with missing data. Machine Learning 45(2), 147–170 (2001)
66. Ferreira da Rocha, J.C., Cozman, F.G.: Inference with separately specified sets of probabilities in credal networks. In: Darwiche, A., Friedman, N. (eds.) Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence (UAI 2002). pp. 430–437. Morgan Kaufmann, San Francisco (2002)
67. Ferreira da Rocha, J.C., Cozman, F.G.: Inference in credal networks with branch-and-bound algorithms. In: Bernard, J.M., Seidenfeld, T., Zaffalon, M. (eds.) ISIPTA. Proceedings in Informatics, vol. 18, pp. 480–493. Carleton Scientific (2003)
68. da Rocha, J.C., Cozman, F.G., de Campos, C.P.: Inference in polytrees with sets of probabilities. In: Conference on Uncertainty in Artificial Intelligence. pp. 217–224. Acapulco (2003)
69. Rubin, D.B.: Inference and missing data. Biometrika 63(3), 581–592 (1976)
70. Takahashi, T.: Debris Flow. A.A. Balkama, Rotterdam (1991), iAHR Monograph
71. Tessem, B.: Interval probability propagation. International Journal of Approximate Reasoning 7(3), 95–120 (1992)
72. Troffaes, M.: Decision making with imprecise probabilities: A short review. In: Cozman, F.G. (ed.) SIPTA Newsletter, pp. 4–7. Society for Imprecise Probability Theory and Applications, Manno, Switzerland (December 2004)
73. Tsoumakas, G., Vlahavas, I.: Random k-Labelsets: An Ensemble Method for Multilabel Classification. In: Proceedings of the 18th European conference on Machine Learning. pp. 406–417. Springer-Verlag Berlin, Heidelberg (2007)
74. Van Der Putten, P., Van Someren, M.: A bias-variance analysis of a real world learning problem: The CoIL challenge 2000. Machine Learning 57(1), 177–195 (2004)
75. Walley, P.: Statistical Reasoning with Imprecise Probabilities. Chapman and Hall, New York (1991)
76. Walley, P.: Inferences from multinomial data: learning about a bag of marbles. J. R. Statist. Soc. B 58(1), 3–57 (1996)
77. Walley, P.: Statistical Reasoning with Imprecise Probabilities, Monographs on Statistics and Applied Probability, vol. 42. Chapman and Hall, London (1991)
78. Walley, P.: Inferences from multinomial data: Learning about a bag of marbles. Journal of the Royal Statistical Society, Series B 58(1), 3–34 (1996)
79. Witten, I., Frank, E.: Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann (2005)
80. Wu, X., Kumar, V., Ross Quinlan, J., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G., Ng, A., Liu, B., Yu, P., et al.: Top 10 algorithms in data mining. Knowledge and Information Systems 14(1), 1–37 (2008)
81. Zaffalon, M.: Statistical inference of the naive credal classifier. In: de Cooman, G., Fine, T.L., Seidenfeld, T. (eds.) ISIPTA '01: Proceedings of the Second International Symposium on Imprecise Probabilities and Their Applications. pp. 384–393. Shaker, The Netherlands (2001)
82. Zaffalon, M.: Conservative rules for predictive inference with incomplete data. In: Cozman, F.G., Nau, R., Seidenfeld, T. (eds.) ISIPTA '05, Proceedings of the Fourth International Symposium on Imprecise Probabilities and Their Applications. pp. 406–415. SIPTA (2005)
83. Zaffalon, M.: Credible classification for environmental problems. Environmental Modelling & Software 20(8), 1003–1012 (2005)
84. Zaffalon, M., Fagiuoli, E.: Tree-based credal networks for classification. Reliable computing 9(6), 487–509 (2003)
85. Zaffalon, M., Hutter, M.: Robust inference of trees. Annals of Mathematics and Artificial Intelligence 45(1), 215–239 (2005)
86. Zaffalon, M., Miranda, E.: Conservative Inference Rule for Uncertain Reasoning under Incompleteness. Journal of Artificial Intelligence Research 34, 757–821 (2009)

87. Zaffalon, M., Wesnes, K., Petrini, O.: Reliable diagnoses of dementia by the naive credal classifier inferred from incomplete cognitive data. Artificial Intelligence in Medicine 29(1-2), 61–79 (2003)