

Tree-Augmented Naive Credal Classifiers

Enrico Fagioli

DISCo, Università degli Studi di Milano-Bicocca
Via Bicocca degli Arcimboldi 8
20126 Milano, Italy
fagioli@disco.unimib.it

Marco Zaffalon

IDSIA
Galleria 2
CH-6928 Manno (Lugano), Switzerland
zaffalon@idsia.ch

Abstract

Credal classification is a generalization of common classification which allows a coherent treatment of imprecision to be realized. This paper defines a credal classifier by extending the discrete tree-augmented naive Bayes classifiers to sets of probability distributions. A solution algorithm is developed whose computational complexity is linear in the number of features. A method to induce the credal classifier from data is proposed.

Keywords: Classification, Naive Bayes classifier, Credal sets, Imprecise probabilities, Bayesian networks.

1 Introduction

Classification is a topic of primary importance for data mining. A classifier is a function that maps instances of a set of variables, called *attributes* or *features*, to a state of a categorical class variable. Such a paradigm is very general: problems in very different fields can be represented as classification problems. Usually, an algorithm called *inducer* builds the classifier from data. The inducer learns by examining examples of attributes-class pairs.

The *naive Bayes classifier* is a simple yet surprisingly accurate model for classification [8, 7, 10]. This classifier is represented in Figure 1 by the graphical language of *Bayesian networks* [13], where the leaves of the graph represent the attributes A_1, \dots, A_n and the root is the class C (each node is also a random variable). The absence of arcs between attributes is equivalent to

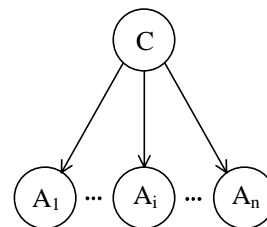


Figure 1: The naive Bayes classifier

state that A_1, \dots, A_n are mutually independent conditional on C .

Tree-augmented naive Bayes classifiers (TANs) are a well-founded and successful extensions of the naive Bayes classifier, obtained by partially relaxing the requirement that features be independent [11]. The dependences between attributes (conditional on C) are modeled by a tree-shaped Bayesian network.

The discrete naive Bayes classifier has also been extended to deal with imprecision in the model probabilities [21]. This has been done by allowing the simultaneous treatment of a set of distributions (namely, a *credal set* [12]) over the variables of the naive Bayes classifier, according to the theory of *imprecise probabilities* [15] (see also [4]). The new model is called *naive credal classifier*. The naive credal classifier can take into account the imprecision in the knowledge extracted from data as generated by small sample sizes and missing data [20], and it does this efficiently and in a theoretical-sound way. As a result, for a given pattern of the attributes, the imprecision in the input may create imprecision in the output. That is, the naive credal classifier recognizes that the available knowledge may not suffice to isolate a

single class, and in this case it provides the user with a set of classes, which are all candidates as the proper class label. In general, a *credal classifier* is defined as a function that maps instances of attributes to a non-empty *subset* of the states of a categorical class variable (a standard classifier is a credal classifier that always outputs singletons).

This paper is concerned with the formal extension of discrete TANs to credal sets and with the development of an algorithm for such new classifiers, which are called *tree-augmented naive credal classifiers* (TANCs). Formally, TANCs are a subset of *credal networks* [9], i.e. Bayesian networks extended to sets of probability distributions. A credal net corresponds to an infinite set of standard Bayesian networks. The computation of a quantity in a credal net involves computing it for each Bayesian network in the set. Usually, one is interested in the extreme values of such a quantity; for this reason, solving global optimization problems becomes a main issue, also considered that they are very often non-linear and non-convex. Not surprisingly, no polynomial exact algorithm of propagation is known even for singly-connected credal nets, and the only general available algorithm can hardly be used for applications [3, 2].

The domain of classification helps solving part of the above complexity, in fact the classification of a complete instance of the attributes does not require propagating credal sets in a TANC, much like in the case of the naive credal classifier (Section 3). When the instance is partially unobserved, computing the TANC classification involves some propagation. We show that TANCs allow such propagation to be realized effectively, by providing a linear-time algorithm. We present such algorithm in Section 3.1, after introducing the notations and some basic concepts in Section 2, and defining credal TANs in Section 3. Section 3.2 discusses the computational complexity of the algorithm, showing that it is linear in the number of nodes of the net. As a result, a complete classification algorithm for TANCs is available, whose computational complexity allows the treatment of real databases. At present, the algorithm learns from complete data sets (Section 4) and credal-classifies possibly incomplete instances. Observe

that it is important to allow values to be missing at least in the instances to classify; in fact, it is generally useful to start getting answers from the classifier also before all the attributes are known.

2 Basic concepts

2.1 Notations

Random variables are represented by capital letters; a random variable and the related node of the graphical model are denoted by the same symbol. For any variable X , let Ω_X ($|\Omega_X| < \infty$) be its frame of discernment. The elements of Ω_X are denoted by lower-case letters (e.g., $X = x$). When no ambiguities arise, the notation $P[X = x | Y = y]$ may be written as $P[x | y]$, for short. The symbol \mathcal{P} is used for a set of probability distributions. When a probability value is interpreted as a function of the unknown distribution P in a set, \underline{P} and \overline{P} denote the minimum and the maximum of the probability, respectively. These are also called *lower and upper probabilities*.

2.2 Credal sets and Bayesian networks

A Bayesian network is a pair $\langle G, P \rangle$ where $G = \langle V, A \rangle$ is a directed acyclic graph, whose nodes (V) are interpreted as variables and whose arcs (A) express the direct dependences between them. Each variable X holds a conditional distribution $P[X | Pa(X)]$ for every possible state $Pa(X)$ of its direct predecessor nodes (or *parents*). It is possible to show [13] that the joint distribution over the variables of the graph is obtained by multiplying the conditional distributions of the nodes: $P[X_1, \dots, X_n] = \prod_{i \in N} P[X_i | Pa(X_i)]$.

The term *credal set* is used for the convex hull of a non-empty and finite set of probability distributions [12]. Geometrically, a credal set is a polytope, i.e. a closed and bounded region which can be described by linear constraints. A credal network [9] is a pair $\langle G, \mathcal{P} \rangle$, where G is like above with the difference that each node X now maintains as many credal sets $\mathcal{P}_X^{Pa(X)}$ of conditional distributions as many joint states $Pa(X)$ of the nodes' parents exist. \mathcal{P} is the set of all the possible joint distributions P over the variables of the net, which is obtained by making every pos-

sible combination of the conditional distributions in the credal sets local to the nodes:

$$\mathcal{P} = \left\{ \begin{array}{l} P[X_1, \dots, X_n] = \prod_{i \in N} P[X_i | Pa(X_i)] : \\ P[X_j | Pa(X_j)] \in \mathcal{P}_{X_j}^{Pa(X_j)}, j = 1 \dots n \end{array} \right\}.$$

Observe that it is possible to give a more general definition of credal network. In fact, here we are assuming that each local credal set of the net can be specified separately from the others, i.e. that the local credal sets are *logically independent* [15, 5]. Relaxing such an assumption has strong negative impact on computational complexity, in a way that also the simplest models become intractable according to current knowledge [18]. On the other hand, violating the above assumption does not produce wrong inferences; in fact it is equivalent to add some further (algorithmic) imprecision, thus making the classifier being more cautious than necessary. Stated differently, this assumption is a way of trading caution for time complexity (see also Section 5).

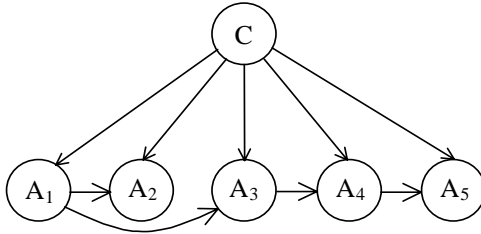


Figure 2: A simple TAN

3 Credal TANs

Figure 2 shows a simple example of a TAN. A TAN is a Bayesian network with nodes C, A_1, \dots, A_n , such that there is an arc from C to any other node, and that the dependencies between attributes form a tree. If the instance to classify is completely known, and we want to compute $P[C | A_1, \dots, A_n]$, each attribute is an evidence node. It is well-known that the arcs leaving evidence nodes can be removed without affecting the computation [13] (recall that the conditional distributions of the attributes must be selected according to the parent's known state). In this case, the computation on the TAN is made as in the case of the naive Bayes classifier. Things are

different when some attributes are not observed. For instance, if attributes A_1, A_3 and A_4 in Figure 2 are missing, the tree between the attributes is unchanged by the presence of some evidence (i.e. A_2 and A_5); therefore the classification depends on a propagation of probability in the tree.

TANCs are special cases of credal nets that have the same graphical structure of TANs. Similarly to the TAN case, the credal-classification of a complete instance (a_1, \dots, a_n) is easy to realize and it is analogous to the case of the naive credal classifier. The naive credal classifier does the classification by tests of credal dominance [21] between each pair of states of C . Consider $c_1, c_2 \in \Omega_C$; c_1 is said to *credal-dominate* c_2 if

$$\min_{P \in \mathcal{P}} P[c_1, a_1, \dots, a_n] - P[c_2, a_1, \dots, a_n] > 0. \quad (1)$$

It is easy to show that c_1 credal-dominates c_2 iff $P[c_1 | a_1, \dots, a_n] > P[c_2 | a_1, \dots, a_n]$ for each $P \in \mathcal{P}$ such that $P[a_1, \dots, a_n] > 0$. The set of classes corresponding to the credal classification is the set of credal-undominated states of C .

Problem (1) can be rewritten as

$$\begin{aligned} \min_{P \in \mathcal{P}_C} \quad & P[c_1] \underline{P}[a_1, \dots, a_n | c_1] - \\ & P[c_2] \overline{P}[a_1, \dots, a_n | c_2] > 0. \end{aligned} \quad (2)$$

Let us examine problem (2). Due to the assumption of separately-specified credal sets, the choice of a distribution $P[C] \in \mathcal{P}_C$ does not affect the choice of other nodes' distributions; hence, the values $P[a_1, \dots, a_n | c_1]$ and $P[a_1, \dots, a_n | c_2]$ can be chosen independently on $P[C]$. Furthermore, these two conditional probabilities are logically independent too: in fact, the credal sets in the dependency tree for the attributes conditional on $C = c_1$ are separated from the credal sets of the tree for $C = c_2$. Then, the choice of the minimum of $P[a_1, \dots, a_n | c_1]$ and the maximum of $P[a_1, \dots, a_n | c_2]$ is a direct consequence, considered that both $P[c_1]$ and $P[c_2]$ are non-negative.

Problem (2) can be further rewritten as follows,

$$\begin{aligned} \min_{P \in \mathcal{P}_C} \quad & P[c_1] \prod_{i=1}^n \underline{P}[a_i | c_1, Pa(A_i)] - \\ & P[c_2] \prod_{i=1}^n \overline{P}[a_i | c_2, Pa(A_i)] > 0, \end{aligned} \quad (3)$$

where $Pa(A_i)$ denotes the instance of the attribute directly preceding A_i that is consistent with a_1, \dots, a_n . Again, the passage to (3) is possible for the separation of the credal sets of different nodes. Notice that problem (2) is a linear program with variables $P[c_1]$ and $P[c_2]$, and as such it is solved efficiently.

Let us now consider an instance for which only the attributes in $\{A_m, \dots, A_n\} = E$ ($1 < m \leq n$) are in a known state, say a_m, \dots, a_n . Testing credal dominance requires the computation of $\min_{P \in \mathcal{P}_C} P[c_1] \underline{P}[a_m, \dots, a_n | c_1] - P[c_2] \overline{P}[a_m, \dots, a_n | c_2]$, following analogous arguments to that leading to (2). But now $\underline{P}[a_m, \dots, a_n | c_1]$ and $\overline{P}[a_m, \dots, a_n | c_2]$ cannot be written as products of known terms, as in (3), since the tree contains some features not in E . The next section develops a procedure that solves this problem by removing the unwanted features from the tree.

3.1 The core algorithm

This section develops a procedure to compute the extremes of $P[a_m, \dots, a_n | c]$ for a given $c \in \Omega_C$. Firstly, the arcs from C to the attributes are removed (in the sequel, we will not consider the conditioning on the class, for sake of clarity, understanding that the computation will be done selecting the local credal sets of the attributes from the original net according to the chosen value of the classification variable). The graph that is produced needs not be connected, i.e. it can be a set of trees, namely a *forest*. We develop the algorithm by assuming that the graph *is* connected. The extension to the case of the forest is described at the end of Section 3.1.2. Secondly, the leaves that do not belong to E are recursively removed. This is actually marginalizing over them, and it is a well-known operation with Bayesian networks. Such operation holds also for credal nets since it holds for all the Bayesian networks that are consistent with the given credal net.

The sections below formally derive a procedure of node absorptions and removals, which is repeatedly applied eventually transforming the tree into a node that represents the state (a_m, \dots, a_n) of the joint variable (A_m, \dots, A_n) .

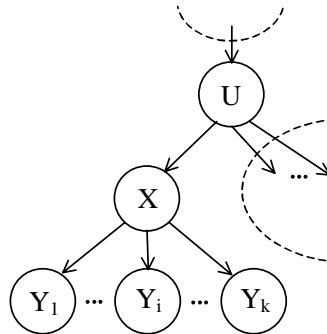


Figure 3: A portion of the tree

3.1.1 Node absorption and removal

We define two operations: absorption and removal. Both apply to a node X whose children, Y_1, \dots, Y_k ($k \geq 1$), are all leaves. Figure 3 shows the portion of tree related to X , its children and its parent U . By construction, Y_1, \dots, Y_k are in E . The operations work under the following conditions: credal sets in different nodes are logically independent. Credal sets in the same node are logically independent when the node is non-leaf. When it is a leaf, a weaker condition suffices. Recall that we are only interested in the instance of the leaves, say y_1, \dots, y_k , which is part of a_m, \dots, a_n and hence the probabilistic information that we need about Y_i ($\forall i \in \{1, \dots, k\}$) reduces to the intervals $[\underline{P}[y_i | x], \overline{P}[y_i | x]]$, $x \in \Omega_X$. For a given $i \in \{1, \dots, k\}$, we require that there exists a distribution $P \in \mathcal{P}$ which attains all the left extremes $\underline{P}[y_i | x]$, $x \in \Omega_X$; similarly for the right extremes (the two distributions need not be the same). We call this condition *consistency of extremes*. Consistency allows us to simultaneously consider either all the left extremes of $P[y_i | x]$, $x \in \Omega_X$, or all the right extremes; but it does not allow us to simultaneously consider, for example, $\underline{P}[y_i | x']$ and $\overline{P}[y_i | x'']$, $x', x'' \in \Omega_X$, $x' \neq x''$, as it would be possible if logical independence was assumed.

Node absorption is used when $X \in E$ (with value $X = x$). Our aim is to cluster node X and all its children into a new node, like in Figure 4, and hence to compute $[\underline{P}[x, y_1, \dots, y_k | u], \overline{P}[x, y_1, \dots, y_k | u]]$ for each $u \in \Omega_U$.

Proposition 1 *The extremes of $P[x, y_1,$*

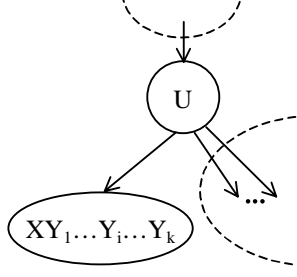


Figure 4: The graph after the absorption of X

$\dots, y_k|u]$ are $\underline{P}[x|u] \prod_{i=1}^k \underline{P}[y_i|x]$ and $\overline{P}[x|u] \prod_{i=1}^k \overline{P}[y_i|x]$, and they are consistent.

Proof. Consider the left extremes (the remaining case is analogous). Let $u \in \Omega_U$; $\underline{P}[x, y_1, \dots, y_k|u] = \underline{P}[x|u] \prod_{i=1}^k \underline{P}[y_i|x]$ follows with analogous arguments to that leading to (3). Given $u', u'' \in \Omega_U$, $u' \neq u''$, $\underline{P}[x|u'] \prod_{i=1}^k \underline{P}[y_i|x]$ is consistent with $\underline{P}[x|u''] \prod_{i=1}^k \underline{P}[y_i|x]$ because in both cases $\underline{P}[y_i|x]$ is set at the same value ($\forall i \in \{1, \dots, k\}$) and because $\underline{P}[x|u']$ and $\underline{P}[x|u'']$ are logically independent. ■

Now we focus on node removal. This is applied when $X \notin E$. We remove node X , we cluster all its children into a new node, like in Figure 5, and we compute $[\underline{P}[y_1, \dots, y_k|u], \overline{P}[y_1, \dots, y_k|u]]$, for each $u \in \Omega_U$.

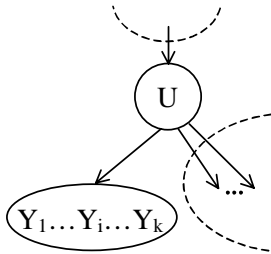


Figure 5: The graph after the removal of X

Proposition 2 *The extremes of $P[y_1, \dots, y_k|u]$ are $\min_{P \in \mathcal{P}_X^u} \sum_x P[x|u] \prod_{i=1}^k \underline{P}[y_i|x]$ and $\max_{P \in \mathcal{P}_X^u} \sum_x P[x|u] \prod_{i=1}^k \overline{P}[y_i|x]$, and they are consistent.*

Proof. Consider $P[y_1, \dots, y_k|u]$, for a given $u \in \Omega_U$. By marginalization and for the independence of Y_1, \dots, Y_k and U , given X , we have $P[y_1, \dots, y_k|u] = \sum_x P[y_1, \dots, y_k|x] P[x|u]$.

This is also $\sum_x P[x|u] \prod_{i=1}^k P[y_i|x]$ because Y_1, \dots, Y_k are mutually independent given X . We take the minimum of the last expression (the maximum is analogous). Recall that the values $\underline{P}[y_i|x]$, $x \in \Omega_X$ are consistent ($\forall i \in \{1, \dots, k\}$) and that credal sets of different nodes are logically independent. Given that all the numbers involved are non-negative, we have:

$$\underline{P}[y_1, \dots, y_k|u] = \min_{P \in \mathcal{P}_X^u} \sum_x P[x|u] \prod_{i=1}^k \underline{P}[y_i|x]. \quad (4)$$

Let us now show that given $u', u'' \in \Omega_U$, $u' \neq u''$, $\underline{P}[y_1, \dots, y_k|u']$ is consistent with $\underline{P}[y_1, \dots, y_k|u'']$. First, observe that the distributions $P[X|u']$ and $P[X|u'']$ are logically independent. Second, $\underline{P}[y_i|x]$ is fixed at the same value in both cases ($\forall i \in \{1, \dots, k\}, \forall x \in \Omega_X$). ■

Observe that both operation can be applied also when X is the root of the tree.

3.1.2 Algorithm

The overall procedure is described in the following points. This takes as input a non-degenerate credal tree (i.e. there must be two nodes at least) whose leaves are all in E .

1. Select a non-leaf node, say X , such that all its children are leaves.
2. If $X \in E$, then absorb it (Proposition 1), else remove it (Proposition 2).
3. If the tree is a single node stop, else go to 1.

Such a procedure computes the wanted extremes. First, trivially, if the tree is not degenerate, there always exists a node like X required by point 1. Second, both operations in point 2 produce a new tree for which: the relevant information concerning the nodes in E is preserved; as far as the credal sets, only a new leaf node is created whose intervals satisfy the consistency property. Therefore, the procedure can iteratively be applied to the new tree. The last operation creates a single node that represents E , so providing us with $[\underline{P}[a_m, \dots, a_n|c], \overline{P}[a_m, \dots, a_n|c]]$.

The extension of the algorithm to a forest is straightforward. The above procedure turns a tree

into a single node. By applying the procedure to each tree, the forest is turned into a set nodes. The nodes are disjoint; for this reason they are probabilistically independent. Further, their related intervals are logically independent by construction. For analogous arguments used to derive (2), $\underline{P}[a_m, \dots, a_n | c]$ is then the product of the lower probabilities of the nodes; similarly for $\overline{P}[a_m, \dots, a_n | c]$.

3.2 Computational complexity

We formalize the computational complexity of a TANC classification with regard to the case of an incomplete instance (the case of completely-known instances is equal to the complexity of the naive credal classifier [21]). We derive the complexity by using a bottom-up approach. Let us start by evaluating the complexity of the operation of node removal.

Expression (4) is a linear program. In order to define such a program, we must compute the inner products. This requires the extremes of $P[y_i | x]$ ($\forall i \in \{1, \dots, k\}, \forall x \in \Omega_X$). We assume that they are already available, as obtained in a pre-processing phase that computes the lower and upper probabilities of the elementary events for all the nodes and all the credal sets. Denote by L the worst-case complexity of solving a linear program in the net, where the worst case is taken over all the polytopes of the net. Let $\overline{\Omega}$ be the maximum number of states of an attribute in the net. (This is also the maximum number of local credal sets of a node, since each node in the tree has at most one parent.) We must compute a linear program (L) for all the states of the nodes ($n\overline{\Omega}$ is the worst-case sum of the states of the nodes) and all the credal sets of each node. This phase takes $O(nL\overline{\Omega}^2)$ and is required to setup the classifier, since the pre-processing is done only once for all the subsequent classifications. For this reason, we do not report it in the following analysis of the complexity of a classification.

Given that the above extremes are available, all the products of the conditional probabilities take $O(od(X)\overline{\Omega})$, denoting by $od(\cdot)$ the outdegree of a node (the number of arcs that leave it). Then we compute expression (4) by means of a linear

program on the probabilities of x given u , thus obtaining $O(od(X)\overline{\Omega} + L)$. Finally, we apply formula (4) for all the states in Ω_U and hence the removal of node X takes $O((od(X)\overline{\Omega} + L)\overline{\Omega})$.

The latter complexity applies to each node X that is removed. Although some nodes are absorbed, we calculate the complexity as if all nodes were removed, thus obtaining an upper bound on the complexity, since the time required for node removal is greater than for node absorption. By summing the last complexity over all the nodes in a tree, we have $O((\overline{\Omega} + L)t\overline{\Omega})$, i.e. the complexity of computing the extremes of the joint probability of E in a tree. This is obtained by considering that L and $\overline{\Omega}$ are constant terms and because the removal of X does not change the outdegree of U . Therefore, the sum of the outdegrees is the sum of the number of arcs, which is bounded by the number of nodes, denoted here by t .

Going back to the decomposition of the original net into a forest, we must sum the latter complexity for each tree in the forest. This gives $O((\overline{\Omega} + L)n\overline{\Omega})$, where n is a bound on the sum of the nodes of the trees (the sum of the t), since the forest is made by a subset of the attributes. Such time is required to compute the extrema of $P[a_m, \dots, a_n | c]$ (Section 3) for a specific class c . Extending it to all the classes, we have $O((\overline{\Omega} + L)n\overline{\Omega}|\Omega_C|)$. Finally, we write the overall complexity of the classification by taking into account the $O(|\Omega_C|^2)$ tests of credal dominance (which are again linear programs; see Section 3), obtaining

$$O(n\overline{\Omega}^2 |\Omega_C| + nL\overline{\Omega} |\Omega_C| + L |\Omega_C|^2). \quad (5)$$

It is useful to specialize this expression to the case of probability intervals, which give rise to an important subset of probability polytopes. Without loss of generality, we consider *reachable* probability intervals (see [1, 21]). In this case, the extremes of the probability of each elementary event are readily available (i.e. no pre-processing is needed). Moreover, the complexity of solving a linear program (L) over reachable probability intervals is linear in the number of the optimization variables [19], and in the case of the tests of credal dominance, the linear program is solved in con-

stant time ([21], Section 4.1). Thus, formula (5) becomes $O\left(n\bar{\Omega}^2 |\Omega_C| + n\bar{\Omega}^2 |\Omega_C| + |\Omega_C|^2\right)$, i.e.

$$O\left(n\bar{\Omega}^2 |\Omega_C| + |\Omega_C|^2\right). \quad (6)$$

Observe that in common applications, both $\bar{\Omega}$ and $|\Omega_C|$ are relatively low. In fact, a necessary condition for Bayesian networks classifiers to be accurate is that the model probabilities have low variance [11, 10]; this would not be met if the above sets were *too* large.

4 Inducing TANCs

The present section discusses the induction of TANCs from a data set in absence of prior knowledge, as it is often the case of data mining applications. We assume that the data set does not contain missing values. In this case, the dependency structure between attributes can be recovered from data according to the standard approach used for TANs [11].

The learning stage is completed with the induction of the nodes' local credal sets. To this extent we propose the method used for the naive credal classifier. This is based on Walley's intervals [16], i.e. lower and upper probabilities for multinomial sampling computed on the basis of an *imprecise* Dirichlet model. For a generic state x of a discrete variable X , the interval is

$$\left[\frac{n_x}{N+s}, \frac{n_x+s}{N+s} \right], \quad (7)$$

where n_x is the number of occurrences of the state x and N is the number of observations. The hyperparameter s reflects the level of caution of the inferences; larger s gives greater caution. Walley provides arguments to choose s in [1, 2]. Intervals (7) correspond to assume a set of Dirichlet priors to represent ignorance. Further, the intervals are naturally reachable, independent on sample space definition, and hence they seem natural candidates for the induction. Computationally, they are fast to calculate, credal sets need little storage to be defined, and the learning is incremental. Finally, since credal sets are defined by reachable probability intervals, the computational complexity of the classification is given by (6). Such complexity allows the expressive power of credal sets to be exploited by TANCs for real applications.

5 Conclusions

The treatment of imprecision is an important step towards the definition of more realistic and reliable models of classification. The theory of imprecise probabilities provides a robust and theoretically-sound framework to this extent. Sets of distributions are greatly expressive, encompassing a number of other models (e.g., possibility measures, belief functions, Choquet capacities, lower probabilities, coherent lower previsions). As a natural consequence, such generality may make things computationally harder. In the authors' experience, this is strongly connected with the treatment of independence. The straight extension of probabilistic independence to sets of distributions requires dealing with (typically difficult) non-linear constraints. More generally, the concept of independence (irrelevance) for credal sets is a very wide and challenging topic [15, 5, 14]. This seems one of the reasons why the joint use of Bayesian networks and credal sets is a field that presents many hard problems [17, 6, 9, 2], but which also seems worth working for.

Among the many possible ways of tackling with the above problems, one is choosing a trade-off between caution and complexity, as proposed in this paper (Section 2.2). Clearly, there must be a proper balance between these two tendencies. The assumption of logically-independent credal sets is much in this direction and it is the only assumption in this paper (the algorithm is exact).

It is important to realize that the degree of caution of the credal classifier may be evaluated experimentally [18]. This is done by comparing the results of the credal classifier and its point-probability counterpart. Thus, experiments may provide precious insights about the choice of the above trade-off.

Acknowledgements

The authors are grateful to an anonymous referee for pointing out an error in the original version of the paper. Marco Zaffalon was partially supported by SUPSI DIE (<http://www.supsi.ch>) under CTI grant # KTI 4217.1.

References

- [1] L. Campos, J. Huete, and S. Moral. Probability intervals: a tool for uncertain reasoning. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 2(2):167–196, 1994.
- [2] A. Cano and S. Moral. A review of propagation algorithms for imprecise probabilities. In G. de Cooman, F. Cozman, S. Moral, and Walley P., editors, *ISIPTA '99*. Univ. of Gent, Belgium: The Imprecise Probabilities Project.
- [3] J.E. Cano, S. Moral, and J.F. Verdegay-López. Propagation of convex sets of probabilities in directed acyclic networks. In B. Bouchon-Meunier, L. Valverde, and R.R. Yager, editors, *Uncertainty in Intelligent Systems*, pages 85–96. Amsterdam: Elsevier, 1993.
- [4] G. de Cooman, P. Walley, and F.G. Cozman, editors. *The Imprecise Probabilities Project*. <http://ippserv.rug.ac.be/>.
- [5] I. Couso, S. Moral, and P. Walley. Examples of independence for imprecise probabilities. In G. de Cooman, F. Cozman, S. Moral, and Walley P., editors, *ISIPTA '99*. Univ. of Gent, Belgium: The Imprecise Probabilities Project.
- [6] F. Cozman. Robustness analysis of Bayesian networks with local convex sets of distributions. In D. Geiger and P.P. Shenoy, editors, *UAI-97*, pages 108–115. San Francisco: Morgan Kaufmann, 1997.
- [7] P. Domingos and M. Pazzani. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning*, 29(2/3):103–130, 1997.
- [8] R.O. Duda and P.E. Hart. *Pattern classification and scene analysis*. New York: Wiley, 1973.
- [9] E. Fagioli and M. Zaffalon. 2U: an exact interval propagation algorithm for polytrees with binary variables. *Artificial Intelligence*, 106(1):77–107, 1998.
- [10] J. Friedman. On bias, variance, 0/1 - loss, and the curse-of-dimensionality. *Data Mining and Knowledge Discovery*, 1:55–77, 1997.
- [11] N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian networks classifiers. *Machine Learning*, 29(2/3):131–163, 1997.
- [12] I. Levi. *The Enterprise of Knowledge*. London: MIT Press, 1980.
- [13] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo: Morgan Kaufmann, 1988.
- [14] P. Vicig. Epistemic independence for imprecise probabilities. In G. de Cooman, F. Cozman, S. Moral, and Walley P., editors, *ISIPTA '99*. Univ. of Gent, Belgium: The Imprecise Probabilities Project.
- [15] P. Walley. *Statistical Reasoning with Imprecise Probabilities*. New York: Chapman and Hall, 1991.
- [16] P. Walley. Inferences from multinomial data: learning about a bag of marbles. *J.R. Statist. Soc. B*, 58(1):3–57, 1996.
- [17] P. Walley. Measures of uncertainty in expert systems. *Artificial Intelligence*, 83:1–58, 1996.
- [18] M. Zaffalon. A credal approach to naive classification. In G. de Cooman, F. Cozman, S. Moral, and Walley P., editors, *ISIPTA '99*. Univ. of Gent, Belgium: The Imprecise Probabilities Project.
- [19] M. Zaffalon. Fast computation of the confidence for the naive credal classifier defined with interval probabilities. Technical Report IDSIA-13-99, IDSIA, 1999. <http://www.idsia.ch/techrep.html>.
- [20] M. Zaffalon. Exact credal treatment of missing data. *Journal of Statistical Planning and Inference*, 2000. To appear.
- [21] M. Zaffalon. The naive credal classifier. *Journal of Statistical Planning and Inference*, 2000. To appear.