Approximating Credal Network Inferences by Linear Programming*

Alessandro Antonucci, Cassio P. de Campos, David Huber and Marco Zaffalon
{alessandro,cassio,david,zaffalon}@idsia.ch

Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA)
Galleria 2, Manno-Lugano (Switzerland)

Abstract. An algorithm for approximate credal network updating is presented. The problem in its general formulation is a multilinear optimization task, which can be linearized by an appropriate rule for fixing all the local models apart from those of a single variable. This simple idea can be iterated and quickly leads to very accurate inferences. The approach can also be specialized to classification with credal networks based on the maximality criterion. A complexity analysis for both the problem and the algorithm is reported together with numerical experiments, which confirm the good performance of the method. While the inner approximation produced by the algorithm gives rise to a classifier which might return a subset of the optimal class set, preliminary empirical results suggest that the accuracy of the optimal class set is seldom affected by the approximate probabilities.

1 Introduction

Credal networks [5] are a generalization of Bayesian networks (e.g., [11]) based on the notion of credal sets. A credal set is a set of probability mass functions, thus representing a quite general and expressive model of uncertainty. Other uncertainty models like belief functions [14] or possibility measures can be regarded as (special classes of) credal sets. A Bayesian network can be turned into a credal network by simply replacing the local models, which are conditional probability mass functions, with conditional credal sets over the same variables. Exactly as a Bayesian network defines a joint probability mass function over its whole set of variables, a credal network defines a joint credal set, which is (the convex closure of) the set of all joint mass functions obtained from the Bayesian networks consistent with the local credal sets.

Compared to the case of Bayesian networks, inference in credal networks is considerably harder. For instance, a marginalization task corresponds to a multilinear optimization problem (updating is a fractional multilinear task) [7]. This is known to be NP-hard even for singly connected networks [8], while the analogous inference in Bayesian networks can be performed in polynomial time [11].

* This work was supported by the Swiss NSF grants nos. 200020_134759 / 1, 200020_137680 / 1, and by the Hasler foundation grant n. 10030.

Despite the hardness of the problem, some algorithms are known to perform reasonably well under certain conditions. Exact approaches have been proposed that implement some branch-and-bound method with local searches [4, 6, 8, 9]. Unfortunately they all suffer from serious efficiency issues unless the credal network is very simple. For instance, none of these methods can deal well with a binary node having four ternary parents, because this setting is already equivalent to $3^4 = 81$ free optimization variables to be chosen, meaning a space of 2^{81} possible solutions just locally to this node! On the other hand, approximate methods either are fast and provide no accuracy guarantee [3, 4, 6] or provide theoretical guarantees but are as slow as exact methods [13]. Moreover, all these approximate methods are only capable of treating credal networks under a vertex-based representation, while a constraint-based specification of credal networks still lacks any practical algorithm.

In this paper we present a fast approximate algorithm for inferences in credal networks based on solving a sequence of linear programming problems. It uses a constraint-based specification, which allows us to deal with domains where the local credal sets are given by their linear constraints. It does not suffer from many parents and large credal sets because the optimization is done by compact linear problems. To the best of our knowledge, this is the first method for general credal networks to truly run the inference with a constraint-based specification. We describe the method and some heuristic ideas to improve its accuracy. Unlike similar ideas already proposed in the literature [6], our approach does not require an explicit enumeration of the extreme points of the credal sets and should be therefore used when the number of extreme points in the local credal sets is exponentially large (e.g., variables with many states and/or parents, credal sets defined by probability intervals, etc). We also discuss how the method can be used for decision making under the maximality criterion [15].

Sections 2 and 3 review the basic notation and definitions of Bayesian and credal networks. The proposed procedure is presented in Sections 4 and 5. Numerical experiments show that the proposed method compares favorably against other available methods in the literature (Section 7). Results are particularly positive when the algorithm is specialized to the case of classification in credal networks based on the maximality criterion. Although this problem is shown to be even harder than the marginalization inferences (discussed in Section 6), classifications based on our approximate algorithm are empirically shown to coincide with those based on exact methods.

2 Bayesian networks

Consider a set of variables $\boldsymbol{X} := (X_0, X_1, \ldots, X_n)$ in one-to-one correspondence with the nodes of an acyclic directed graph $\mathcal{G}$. For each $i = 0, \ldots, n$, the joint variable $\Pi_i \subseteq \boldsymbol{X}$ denotes the parents of X_i according to $\mathcal{G}$. All these variables are categorical: X_i takes its values on the finite set Ω_{X_i} and so does Π_i in $\Omega_{\Pi_i} := \times_{X_j \in \Pi_i} \Omega_{X_j}$, for each $i = 0, \ldots, n$.[1] The graph $\mathcal{G}$ represents stochastic

[1] Symbol $\times$ denotes Cartesian set product.

independence relations by means of a Markov condition: any variable is conditionally independent of its non-descendant non-parents given its parents (see e.g., [11]). The specification of a conditional probability mass function $P(X_i|\pi_i)$ for each $\pi_i \in \Omega_{\Pi_i}$ and $i = 0, \ldots, n$, induces, for each $\boldsymbol{x} \in \times_{i=0}^{n} \Omega_{X_i}$, the factorization:

$$P(\boldsymbol{x}) := \prod_{i=0}^{n} P(x_i|\pi_i), \tag{1}$$

where the values of x_i and π_i are those consistent with $\boldsymbol{x}$.

We call *Bayesian network* a specification of the conditional probability mass functions $\{P(X_i|\pi_i)\}_{i=0,\ldots,n}^{\pi_i \in \Omega_{\Pi_i}}$. In particular, the mass functions associated to X_i, i.e., $\{P(X_i|\pi_i)\}_{\pi_i \in \Omega_{\Pi_i}}$ are called the *local models* of X_i, for each $i = 0, \ldots, n$. Inference in Bayesian networks is based on the joint probability mass function in Eq. (1). Marginals, for instance, are obtained by summing out other variables from the joint, i.e., the marginalization of X_0 corresponds to the computation, for each $x_0 \in \Omega_{X_0}$, of

$$P(x_0) = \sum_{x_1,\ldots,x_n} \prod_{i=0}^{n} P(x_i|\pi_i), \tag{2}$$

where $\sum_x$ is a shortcut for $\sum_{x \in \Omega_X}$. With straightforward calculations, the marginal in Eq. (2) can be expressed as a linear combination of the local probabilities associated to an arbitrary $X_j \in \boldsymbol{X}$, i.e.,

$$P(x_0) = \sum_{x_j, \pi_j} [P(x_0|x_j, \pi_j) \cdot P(\pi_j)] \cdot P(x_j|\pi_j), \tag{3}$$

where probabilities $P(\pi_j)$ and $P(x_0|x_j, \pi_j)$ can be computed from the joint as in Eq. (1),[2] while probabilities $P(x_j|\pi_j)$ are already available in the Bayesian network specification. As special case, note that for $j = 0$, Eq. (3) rewrites as $P(x_0) = \sum_{\pi_0} P(\pi_0) \cdot P(x_0|\pi_0)$; while if $X_0 \in \Pi_j$, and $\Pi'_j := \Pi_j \setminus \{X_0\}$, $P(x_0) = \sum_{x_j, \pi'_j} P(x_0, \pi'_j) P(x_j|x_0, \pi'_j)$. Remarkably, values of both $P(\pi_j)$ and $P(x_0|x_j, \pi_j)$ are not affected by those of the local models of X_j in the Bayesian network specification. To see that, note that when computing a marginal, the descendants and hence their local models can be removed without affecting the probability. As X_j is a child of all the variables in Π_j, the computation of $P(\pi_j)$ is not affected by the local models $\{P(X_j|\pi_j)\}_{\pi_j \in \Omega_{\Pi_j}}$. Similarly, when computing a conditional probability, arcs leaving the variables after the conditioning bar can be removed: thus, in the case of $P(x_0|x_j, \pi_j)$, we can disconnect X_j from the rest of the network, thus making its local model irrelevant for the particular calculation. This remark, together with Eq. (3) will be exploited by the approximate algorithm presented later.

[2] Given a joint probability mass function, conditionals are obtained from Bayes' rule. For instance, $P(x_0|x_j, \pi_j) = P(x_0, x_j, \pi_j)/P(x_j, \pi_j)$, provided that $P(x_j, \pi_j) > 0$.

3 Credal networks

The Bayesian theory of subjective probability has been extended by more general uncertainty theories in order to model situations of highly incomplete or conflicting information. Among others, the theory of imprecise probability in [15] adopts credal sets, which are (convex) sets of probability mass functions, as a more general model of uncertainty about the state of a categorical variable. In particular, here we focus on finitely generated credal sets, which are specified by a finite number of linear constraints on the probabilities (e.g., see Fig. 1).

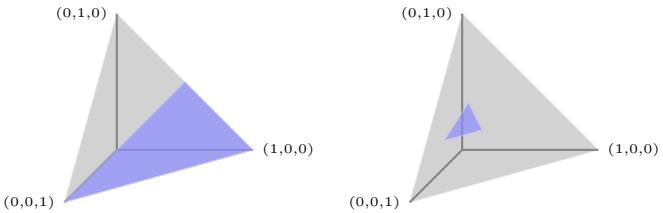

Fig. 1. Credal sets over a ternary variable X (i.e., $\Omega_X = \{x', x'', x'''\}$). The representation is in a three-dimensional space with coordinates $[P(x'), P(x''), P(x''')]$. The polytopes represent respectively: (a) the credal set defined by constraint $P(x') > P(x'')$; (b) a credal set whose extreme points are $\{[.1, .3, .6]^T, [.3, .3, .4]^T, [.1, .5, .4]^T\}$.

Credal sets can be used to extend Bayesian networks to imprecise probabilities. In order to do that, in the definition of Bayesian network, every conditional probability mass function $P(X_i|\pi_i)$ is replaced by a (conditional) *credal set* $K(X_i|\pi_i)$ for each $\pi_i \in \Omega_{\Pi_i}$ and $i = 0, \ldots, n$. As we focus on credal sets defined by a finite number of linear constraints, the set of extreme points of $K(X_i|\pi_i)$, to be denoted by $\text{ext}[K(X_i|\pi_i)]$, has finite cardinality. We call *credal network* a specification of conditional credal sets $\{K(X_i|\pi_i)\}_{i=0,\ldots,n}^{\pi_i \in \Omega_{\Pi_i}}$. Under this generalized setting, Eq. (1) can be used to obtain different joint probability mass functions. Let us consider all the possible extreme specifications, and then take the convex hull (denoted as CH), i.e., build the following joint credal set:

$$K(\boldsymbol{X}) := \text{CH}\left\{ P(\boldsymbol{X}) \,\middle|\, P(\boldsymbol{x}) := \prod_{i=0}^{n} P(x_i|\pi_i), \begin{array}{l} \forall \boldsymbol{x} \in \times_{i=0}^{n} \Omega_{X_i}, \\ \forall P(X_i|\pi_i) \in \text{ext}[K(X_i|\pi_i)] \\ \forall i = 0, 1, \ldots, n, \forall \pi_i \in \Omega_{\Pi_i} \end{array} \right\}. \tag{4}$$

The credal set in Eq. (4) is called the *strong extension* of the credal network. Here inference in credal networks is intended as based on the strong extension. For instance, the lower bound with respect to $K(\boldsymbol{X})$ of the marginal probability

in Eq. (2) is:

$$\underline{P}(x_0) := \min_{P(\boldsymbol{X})\in K(\boldsymbol{X})} P(x_0) = \min_{\substack{P(X_i|\pi_i)\in K(X_i|\pi_i)\\ \pi_i\in\Omega_{\Pi_i}, i=0,\ldots,n}} \sum_{x_1,x_2,\ldots,x_n} \prod_{i=0}^{n} P(x_i|\pi_i), \qquad (5)$$

and similarly for the upper $\overline{P}(x_0)$. Eq. (5) corresponds to the optimization of a non-linear (namely multilinear [7]) function over a feasible region defined by linear constraints on the optimization variables. In the next section we present an approximate algorithm for this task.

4 The algorithm

The algorithm we present is based on Lukatskii and Shapot's approach [12] to approximate the solution of multilinear problems. In essence, a multilinear problem can be converted into a linear one if we fix all but one optimization variable in each of its multilinear terms. In Lukatskii and Shapot's terminology, there is a partition $S_1 \cup S_2 \cup \ldots \cup S_w$ of the optimization variables such that fixing the optimization variables in every set of the partition apart from S_j, the multilinear problem becomes linear. By iterating over j, which defines the set S_j to remain free, one can approximate the solution of the multilinear problem with a sequence of linear ones. Da Rocha et al. [10] have already used similar ideas to perform approximate inference in credal networks, but their approach had to enumerate all the extreme points of credal sets and used a less sophisticated search.

Our algorithm finds an inner approximation of the interval $[\underline{P}(x_0), \overline{P}(x_0)]$, i.e., an upper approximation of the lower probability as in Eq. (5) and a lower approximation of the upper probability. The idea is to reduce the multilinear task in Eq. (5) to a linear program by fixing all the local credal sets to singletons apart from those associated to an arbitrarily chosen variable $X_j \in \boldsymbol{X}$, which we call the *free* variable. Given a free $X_j \in \boldsymbol{X}$, we pick an extreme point $\tilde{P}(X_i|\pi_i) \in \text{ext}[K(X_i|\pi_i)]$, for each $\pi_i \in \Omega_{X_i}$ and $i = 0, \ldots, n$, $i \neq j$. These are additional constraints to the optimization problem in Eq. (5), which becomes:

$$\begin{aligned}\underline{P}'(x_0) &:= \min_{\substack{P(X_j|\pi_i)\in K(X_j|\pi_j)\\ \pi_j\in\Omega_{\Pi_j}}} \sum_{x_1,x_2,\ldots,x_n} \left[\prod_{i=0, i\neq j}^{n} \tilde{P}(x_i|\pi_i)\right] \cdot P(x_j|\pi_j) = \\ &= \min_{P(X_j|\pi_j)\in K(X_j|\pi_j)} \sum_{x_j,\pi_j} \left[\tilde{P}(x_0|x_j,\pi_j)\cdot\tilde{P}(\pi_j)\right]\cdot P(x_j|\pi_j),\end{aligned} \qquad (6)$$

where the last derivation is based on Eq. (3) and probabilities $\tilde{P}(x_0|x_j,\pi_j)$ and $\tilde{P}(\pi_j)$ are denoted by a tilde as they are computed from the joint of a Bayesian network with local models $\{\tilde{P}(X_i|\pi_i)\}$. The discussion of the special cases $j = 0$ and $X_0 \in \Pi_j$ is exactly as in the Bayesian case (see the end of Sect. 2). We focus on marginal probabilities just for the sake of clarity. Yet, the computation

with conditional probabilities is straightforward as the linear programs become linear-fractional programs.

Let us comment on two important facts about Eq. (6). First, being the solution of an optimization with additional constraints with respect to Eq. (5) (see the second term in the equation), we clearly have $\underline{P}(x_0) \leq \underline{P}'(x_0)$. Secondly, it is clear from the third term of Eq. (6) that the computation of $\underline{P}'(x_0)$ is a linear program whose optimization variables are the local probabilities of X_j, i.e., $\{P(x_j|\pi_j)\}_{x_j \in \Omega_{X_j}, \pi_j \in \Omega_{\Pi_j}}$. Moreover, as the solution of a linear program lies on an extreme point of the feasible region (i.e., an extreme point of the credal set), there is a specification $P^*(X_j|\pi_j) \in \text{ext}[K(X_j|\pi_j)]$, for each $\pi_j \in \Omega_{\Pi_j}$ such that:

$$\underline{P}'(x_0) = \sum_{x_1, x_2, \ldots, x_n} P^*(x_j|\pi_j) \left[\prod_{i=0, i \neq j}^{n} \tilde{P}(x_i|\pi_i) \right]. \tag{7}$$

Coping with zero probabilities. In order to obtain the coefficients of the objective function in the linear task in Eq. (6), the conditionals $\tilde{P}(x_0|x_j, \pi_j)$ (and the marginals $\tilde{P}(\pi_j)$) should be computed for each $x_j \in \Omega_{X_j}$, $\pi_j \in \Omega_{\Pi_j}$. For zero-probability conditioning events, i.e., $\tilde{P}(x_j, \pi_j) = 0$, the conditionals cannot be computed. In this case, the term of the sum in Eq. (6) associated to (x_j, π_j) rewrites as: $\tilde{P}(x_0|x_j, \pi_j) \cdot \tilde{P}(\pi_j) \cdot \tilde{P}(x_j|\pi_j) = \tilde{P}(x_0|x_j, \pi_j) \cdot \tilde{P}(x_j, \pi_j)$, being therefore zero. Thus, the corresponding term does not appear in the objective function and its coefficient can be safely set to zero.

5 Searching for the optimum

In the previous section, we defined a procedure which, given a free variable X_j and the specification of an extreme point for all conditional credal sets of non-free variables, returns an upper approximation of the lower probability $\underline{P}(x_0)$, together with the specification of the extreme points of the local credal sets associated to the free variable which produced that optimum.

If we call *almost-Bayesian network* a credal network whose local credal sets are singletons apart from those associated to a single variable, the optimization procedure we proposed consists in taking an almost-Bayesian network consistent with the original credal network (i.e., its strong extension is included in that of the original credal network) and exploiting the fact that marginalization of almost-Bayesian networks is a linear problem. By solving the linear problem, we obtain: (i) an upper (lower) approximation of the lower (upper) probability; (ii) a specification of the extreme points of the credal sets associated to the only "non-Bayesian" variable in the almost-Bayesian network. These extreme points can be used as an assignment for the extreme points of those local credal sets, and another variable can be "freed", leading to a new linear program. In the rest of this section we suggest a possible initialization and two iteration strategies.

Initialization. The optimization in Eq. (6) requires an initialization, i.e., the specification of an almost-Bayesian network consistent with the credal network. This can be done by randomly picking an extreme point (or a simple point) from each local credal set apart for those associated to X_j. A deterministic alternative to the random choice is the center of mass of the credal set:[3]

$$P_{\mathrm{CM}}(x_i|\pi_i) := \sum_{P(X_i|\pi_i)\in \mathrm{ext}[K(X_i|\pi_i)]} \frac{P(x_i|\pi_i)}{|\mathrm{ext}[K(X_i|\pi_i)]|} \tag{8}$$

for each $x_i \in \Omega_{X_i}$, $\pi_i \in \Omega_{\Pi_i}$, $i = 0, 1, \ldots, n$, with $i \neq j$. Note that the center of mass belongs to its credal set, but it is not an extreme point of it (unless the credal set includes a single point). As we know that the exact solution of Eq. (5) corresponds to a Bayesian network whose local models are extreme points of the local credal sets, this means that if a solution includes a center of mass it cannot be exact. Yet, this can be easily overcome by iterating the procedure at least once for each variable, as all those linear problems will certainly pick extreme points.

Greedy search The solution in Eq. (7) of the linear program in Eq. (6) provides an approximate solution for the computation of the marginal of a credal network. This procedure can be iterated by changing the "free" variable X_j and using the optimal solution $\{P^*(X_j|\pi_j)\}_{\pi_j\in\Omega_{\Pi_j}}$ of the previous problem as a different initialization. This improves the solution as shown here.

Proposition 1. *Let $\{\tilde{P}(X_j|\pi_j)\}_{j=0,1,\ldots,n}^{\pi_j\in\Omega_{\Pi_j}}$ be a Bayesian network specification consistent with a credal network specification $\{K(X_j|\pi_j)\}_{j=0,1,\ldots,n}^{\pi_j\in\Omega_{\Pi_j}}$. As in Eq. (2), let $\tilde{P}(x_0) := \sum_{x_1,\ldots,x_n}\prod_{i=0}^{n}\tilde{P}(x_i|\pi_i)$ and, as in Eq. (6):*

$$\underline{\tilde{P}}'(x_0) := \min_{P(X_j|\pi_j)\in K(X_j|\pi_j)} \sum_{x_1,\ldots,x_n} \left[\prod_{\substack{i=0\\ i\neq j}}^{n} \tilde{P}(x_i|\pi_i)\right] P(x_j|\pi_j). \tag{9}$$

Then $\underline{\tilde{P}}'(x_0) \leq \tilde{P}(x_0)$.

Proof. It suffices to put in evidence the terms $\{\tilde{P}(x_j|\pi_j)\}^{\pi_j\in\Omega_{\Pi_i}}$ in the definition of $\tilde{P}(x_0)$ and note that, by definition of consistency between Bayesian and credal networks, $\tilde{P}(X_j|\pi_j) \in K(X_j|\pi_j)$ for each $\pi_j \in \Omega_{\Pi_j}$. □

As a corollary of Prop. 1, it follows that iterating the algorithm can only improve the quality of the approximation. A *greedy* iteration strategy is therefore the following: given a candidate solution $\underline{P}(x_0)$, we evaluate the improved solution obtained by keeping the same specification of the extreme mass functions as those used to obtain $\underline{P}(x_0)$ and we free one of the variables a time. Let $\underline{P}'_j(x_0)$

[3] In the language of evidence theory [14], this corresponds to the so-called *pignistic* transformation which associates a probability mass function to a belief functions.

denote the candidate solution obtained by freeing X_j, for each $j = 0, 1, \ldots, n$. During the first iteration, we pick as free variable X_{j^*} such that:

$$j^* := \text{argmin}_{j=0,1,\ldots,n} \underline{P}'_j(x_0). \tag{10}$$

This naturally provides us with a partition of the optimization variables as defined by Lukatskii and Shapot [12]. Hence, if all estimated solutions in Eq. (10) coincide with the previously obtained solution, a stationarity area has been reached and the algorithm stops. Often this will be a local optimum of the multilinear problem. Yet, this is not always the case because there might be a neighborhood of candidates with no improving solution, but whose neighbors might have an improving solution. The only way to ensure local optimality is to keep track of all the candidates with equal solution until such set is completely explored or an improving solution is found [12]. In practice, this is not an issue, and can be overcome by the use of multiple starts, perturbations of solutions in case of achieving a stationary area, and/or a queue of candidate solutions, as we describe in the following.

Improving the greedy approach. The greedy approach based on Eq. (10) and described in the previous paragraph can be improved by defining a *priority queue* of size k, which includes not only the best candidate, but the k-best ones (each candidate is tracked together with its relative Bayesian network specification). The solutions $\{\underline{P}_j(x_0)\}_{j=0}^n$ are evaluated for the candidate in the peak of the priority queue, and are themselves included back in the queue (as long as they are improving solutions). In this variant, the algorithm stops when the queue is empty, which guarantees that all candidates have been explored (this will certainly include the previously explained greedy approach). The queue can be seen as many greedy searches in distinct "directions".

Computational complexity (algorithm). Let m and l denote, respectively, the maximum number of states and incoming parents (i.e., the indegree) of the network variables: $m := \max_{i=0,\ldots,n} |\Omega_{X_i}|$ and $l := \max_{i=0,\ldots,n} |\Pi_i|$. Let q be the maximum number of linear constraints required to define a local credal set. A linear program as in Eq. (5) has at most m^{l+1} variables and $m^l \cdot q$ constraints. Because the input size should already be proportional to $m^l \cdot q$, the algorithm spends time equivalent to run a linear programming solver on the (local) input specification times the total number of iterations.

6 Maximality-based classification

Credal networks have been used to implement both knowledge-based systems (e.g., [1]) and classifiers (e.g., [16]). Given a credal network over $\boldsymbol{X}$, let X_0 be the class variable and $\tilde{\boldsymbol{X}} \subseteq \boldsymbol{X} \setminus \{X_0\}$ the variables (features) for which evidential information is available. Given an instance $\tilde{\boldsymbol{x}}$ of the features, the identification of the optimal class(es) of X_0 should be therefore based on the conditional credal set $K(X_0|\tilde{\boldsymbol{x}})$ obtained by conditioning the strong extension in Eq. (4). Such an

identification depends on the adopted decision criterion. E.g., the so called Γ-maximin approach returns $x_0^* := \text{argmax}_{x_0 \in \Omega_{X_0}} \underline{P}(x_0|\tilde{\boldsymbol{x}})$. Another criterion is maximality [15], which returns the following set of classes:

$$\Omega^*_{X_0} := \left\{ x'_0 \in \Omega_{X_0} \;\middle|\; \nexists x''_0 \in \Omega_{X_0} : \begin{array}{l} P(x''_0|\tilde{\boldsymbol{x}}) > P(x'_0|\tilde{\boldsymbol{x}}) \\ \forall P(X_0|\tilde{\boldsymbol{x}}) \in K(X_0|\tilde{\boldsymbol{x}}) \end{array} \right\}. \tag{11}$$

In practice, $\Omega^*_{X_0}$ should be initialized to Ω_{X_0}. Then, for each $x'_0, x''_0 \in \Omega_{X_0}$, the following dominance should be checked:

$$\min_{P(X_0|\tilde{\boldsymbol{x}}) \in K(X_0|\tilde{\boldsymbol{x}})} [P(x''_0|\tilde{\boldsymbol{x}}) - P(x'_0|\tilde{\boldsymbol{x}})] > 0, \tag{12}$$

and, if satisfied, x'_0 removed from $\Omega^*_{X_0}$. The test in Eq. (12) cannot be directly checked by algorithms for credal networks. Nevertheless, in a recent paper [2], the test has been mapped to a standard updating task in a credal network. This is obtained by augmenting the original credal network with an auxiliary node associated to a Boolean variable Y and such that Y is a leaf child of X_0. The quantification of the conditional credal sets for Y given X_0 is precise:

$$P(Y = \text{true}|x_0) = \begin{cases} 0 \text{ if } & x_0 = x'_0 \\ 1 \text{ if } & x_0 = x''_0 \\ \frac{1}{2} \text{ otherwise.} & \end{cases} \tag{13}$$

After this quantification, the dominance test in Eq. (12) is equivalent to check whether $\underline{P}(Y = \text{true}|\tilde{\boldsymbol{x}}) > \frac{1}{2}$. The algorithm proposed in [2] can be used to evaluate the dominance for each pair of classes and determine the undominated ones according to Eq. (11). The upper approximation $\underline{P}'(Y = \text{true}|\tilde{\boldsymbol{x}}) \geq \underline{P}(Y = \text{true}|\tilde{\boldsymbol{x}})$ implies that some dominances detected by the algorithm might not really take place. Hence, the set of optimal classes evaluated by the approximate algorithm is a subset of the exact one.

Computational complexity (classification). We characterize the computational complexity of maximality-based classification. The evaluation in Eq. (12) is called *dominance test*. Given a credal network, evidence $\tilde{\mathbf{x}}$, $q \in \Omega_Q$, and a rational r, the *inference query* decides whether exists $P \in K(\mathbf{X})$ such that $P(q|\tilde{\mathbf{x}}) \geq r$ [8]. The treewidth of a network measures the extent to which it resembles a tree (see [11] for a more formal definition).

Theorem 1. *The dominance test is coNP-complete in bounded treewidth networks and coNP^{PP}-complete in networks of general topology.*

Proof. We show hardness by demonstrating that the complementary decision, that is, whether the minimization of Eq. (12) *is less than or equal to zero, is NP^{PP}-hard in general, and NP-hard for bounded treewidth networks. For that, we reduce the marginal inference problem in a credal network to it. Marginal inference in credal networks is shown to be NP-hard in polytrees with at most two parents per node and NP^{PP}-hard in general networks [8].*

Take a credal network with inference query $\exists P : P(q|\tilde{\mathbf{x}}) \geq r$, for a given rational r, query q and evidence $\tilde{\mathbf{x}}$. Build a new network by adding a binary node X_0, which has Q as sole parent and precise probability mass functions defined as $P(x_0''|q) = \frac{r}{2}$ and $P(x_0''|\neg q) = \frac{1+r}{2}$. Note that the new network has the same topology (and treewidth) of the original one. Now, the complement of the dominance test asks whether

$$\min_P [P(x_0''|\tilde{\mathbf{x}}) - P(x_0'|\tilde{\mathbf{x}})] \leq 0 \iff \min_P [2P(x_0''|\tilde{\mathbf{x}}) - 1] \leq 0$$

$$\iff \min_P [rP(q|\tilde{\mathbf{x}}) + (1+r)P(\neg q|\tilde{\mathbf{x}})) - 1] \leq 0 \iff$$

$$\min_P [r - P(q|\tilde{\mathbf{x}})] \leq 0 \iff \max_P P(q|\tilde{\mathbf{x}}) \geq r \iff \exists P : \ P(q|\tilde{\mathbf{x}}) \geq r,$$

which is exactly the credal network marginal query. As the treewidth of the network has not been modified, the hardness results follow. Pertinence of this complementary decision in NP for the case of bounded treewidth (respectively in NP^{PP} for the general case) is immediate, since given $P \in K(\mathbf{X})$, we can use a Bayesian network inference to certify that $P(x_0''|\tilde{\mathbf{x}}) \leq P(x_0'|\tilde{\mathbf{x}})$ (in polynomial time for bounded treewidth nets and by using the PP oracle for the general case). □

Hence, deciding whether a class is in the maximal set is a very demanding query, because it is tested against all other classes, and each of such tests can be itself hard. For example, if the network has bounded treewidth, the problem of deciding whether a class is maximal falls in the class of decision problems that can be solved by polynomial time machines with access to non-adaptive queries to an NP oracle, namely $\mathrm{P}^{||\mathrm{NP}}$. Even if a hard task to do exactly, we shall see that our algorithm is able to recover the set of maximal classes successfully in practice (but without guarantee of exactness).

7 Experiments

To validate the performance of our algorithm, we use a benchmark made of different credal nets with random topology, either multiply or singly connected, and two classical (multiply connected) models, namely the *Alarm* and the *Insurance* networks. The maximum indegree for the networks with random topology is limited to 5. The number of states for the Alarm and the Insurance networks is the same as in their original specifications, while for the other networks the number of states is randomly chosen between 2 and 8. All the models are quantified by randomly generated conditional credal sets with a fixed number of extreme points, whose number is ranging from 2 to 8 for each network. Inferences are computed by a Java implementation of the algorithm linked to the COIN-OR linear program solver. The code is available as a free software tool.[4] In these experiments, the greedy approach described in Sect. 4 is considered and the algorithm is therefore called G-LP. Centers of mass are used for the first iteration.

[4] See `http://ipg.idsia.ch/software` and `http://www.coin-or.org`.

Exact inferences are computed by mapping the problem to an integer linear program [9], which is solved by CPLEX. Comparisons are with other approximate algorithms: the iterated local search (ILS) [6] and the GL2U algorithm [3].

Networks	# of tests	G-LP	G-LP'	GL2U	ILS
Alarm	973	.0474	.0076	.1218	.2709
Insurance	650	.0767	.0795	.1818	.2700
Random (single)	6162	.0816	.0109	.1724	.1528
Random (multi)	2963	.0855	.0140	.1594	.1269

Table 1. Benchmark results (mean square absolute errors). Upper marginal probabilities have been computed for each state of each network in the benchmark such that the exact solver took less than three minutes to find the optimum.

Before commenting on the results in Tab. 1, note that our approach assumes the local credal sets to be specified by linear constraints. This is often the case in real scenarios (e.g., credal classifiers or knowledge-based expert systems quantified by probability intervals). Conversely, credal networks used for benchmarking represent their local credal sets by explicit enumeration of the extreme points. The reason is that most of the algorithms for credal networks require the local credal sets to be described by their extreme points. In the first experiment, we evaluate the lower and upper bounds of the probabilities w.r.t. the extreme points. E.g., for the credal set in Fig. 1(b): $P(x') \in [.1, .3]$, $P(x'') \in [.3, .5]$, $P(x''') \in [.4, .6]$. These constraints define larger credal sets compared to the original ones. The third column of Tab. 1 reports the results, i.e., the mean square difference between the inner approximation obtained by G-LP and the exact inferences. The accuracy is fairly good on the whole benchmark. In the fourth column of Tab. 1, the same inferences computed by G-LP in the third column are regarded here as approximations for credal networks with local credal sets defined by the original extremes (and not by the induced linear constraints). We denote this heuristic variant as G-LP'. The inner approximation of G-LP is now balanced by the outer approximation introduced by considering the linear constraints and this produces smaller errors (MSE$<$ .02) for the Alarm and the random networks. The performance is less accurate for the Insurance network, probably because of the relatively high number of states for the variables of this network, which makes the outer approximation too coarse. Regarding the proposed improvement of the greedy approach, the results (with queue size $k = 40$ and still one minute as maximum running time) are just marginally better than those based on G-LP and for this reason are not reported. Finally, we evaluate the performance of G-LP for maximality-based classification. We consider ten benchmark networks with the Alarm topology. As classes we choose the variables with four states with no evidence (i.e., $\tilde{\boldsymbol{X}} = \emptyset$). On *all* these classification tasks the two sets of optimal classes coincide. Thus, the small approximation in the inferences based on G-LP seems to have no effect when finding the set of maximal solutions, which were recovered exactly. While this is somehow expected (because the dependency on the exact probability value is less important), this empirical result is promising for the use of credal networks in classification.

8 Conclusions

A new algorithm based on a sequence of linear optimizations is proposed for approximate credal network updating. The algorithm can deal with a constraint-based specification of credal networks, and provides inner approximation solutions. It is also extended to find the maximal classes in a classification problem. The complexities of these problems and of the algorithm are presented. In a practical perspective, preliminary results are promising: the algorithm is fast and accurate. As future work, we intend to test the algorithm on larger networks and with other search heuristics, and support other decision criteria.

References

1. A. Antonucci, R. Brühlmann, A. Piatti, and M. Zaffalon. Credal networks for military identification problems. *Int. J. Approx. Reasoning.*, 50(2):666–679, 2009.
2. A. Antonucci and C.P. de Campos. Decision making by credal nets. In *Proceedings of the International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC 2011)*, volume 1, pages 201–204. IEEE, 2011.
3. A. Antonucci, S. Yi, C.P. de Campos, and M. Zaffalon. Generalized loopy 2U: a new algorithm for approximate inference in credal networks. *Int. J. Approx. Reasoning.*, 51(5):474–484, 2010.
4. A. Cano, M. Gomez, S. Moral, and J. Abellan. Hill-climbing and branch-and-bound algorithms for exact and approximate inference in credal networks. *Int. J. Approx. Reasoning.*, 44(3):261–280, 2007.
5. F. G. Cozman. Credal networks. *Artificial Intelligence*, 120:199–233, 2000.
6. J. C. F. da Rocha, F. G. Cozman, and C. P. de Campos. Inference in polytrees with sets of probabilities. In *UAI '03*, pages 217–224, 2003.
7. C. P. de Campos and F. G. Cozman. Inference in credal networks using multilinear programming. In *Proceedings of the Second Starting AI Researcher Symposium*, pages 50–61, Amsterdam, 2004. IOS Press.
8. C. P. de Campos and F. G. Cozman. The inferential complexity of Bayesian and credal networks. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1313–1318, Edinburgh, 2005.
9. C. P. de Campos and F. G. Cozman. Inference in credal networks through integer programming. In *International Symposium on Imprecise Probability: Theories and Applications (ISIPTA)*, pages 145–154, Prague, 2007.
10. J.C. Ferreira da Rocha and F.G. Cozman. Inference in credal networks: branch-and-bound methods and the A/R+ algorithm. *Int. J. Approx. Reasoning.*, 39(2-3):279–296, 2005.
11. D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.
12. A. M. Lukatskii and D. V. Shapot. Problems in multilinear programming. *Computational Mathematics and Mathematical Physics*, 41(5):638–648, 2000.
13. D.D. Mauà, C.P. de Campos, and M. Zaffalon. Updating credal networks is approximable in polynomial time. *Int. J. Approx. Reasoning.*, 2012.
14. G. Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.
15. P. Walley. *Statistical Reasoning with Imprecise Probabilities*. Chapman and Hall, 1991.
16. M. Zaffalon. The naive credal classifier. *J. Stat. Plann. Inference*, 105(1):5–21, 2002.