Abductive Reasoning in Logical Credal Networks

Radu Marinescu
IBM Research, Ireland
radu.marinescu@ie.ibm.com

Junkyu Lee
IBM Research, USA
junkyu.lee@ibm.com

Debarun Bhattacharjya
IBM Research, USA
debarunb@us.ibm.com

Fabio Cozman
Universidade de São Paulo, Brazil
fgcozman@usp.br

Alexander Gray
Centaur AI Institute, USA
alexander.gray@centaurinstitute.org

Abstract

Logical Credal Networks or LCNs were recently introduced as a powerful probabilistic logic framework for representing and reasoning with imprecise knowledge. Unlike many existing formalisms, LCNs have the ability to represent cycles and allow specifying marginal and conditional probability bounds on logic formulae which may be important in many realistic scenarios. Previous work on LCNs has focused exclusively on marginal inference, namely computing posterior lower and upper probability bounds on a query formula. In this paper, we explore abductive reasoning tasks such as solving MAP and Marginal MAP queries in LCNs given some evidence. We first formally define the MAP and Marginal MAP tasks for LCNs and subsequently show how to solve these tasks exactly using search-based approaches. We then propose several approximate schemes that allow us to scale MAP and Marginal MAP inference to larger problem instances. An extensive empirical evaluation demonstrates the effectiveness of our algorithms on both random LCN instances as well as LCNs derived from more realistic use-cases.

1 Introduction

Probabilistic logic which combines probability and logic in a principled manner has emerged over the past decades as a unified representational and reasoning framework capable of dealing effectively with complex real-world applications that require efficient handling of uncertainty and compact representations of domain expert knowledge [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]. Logical Credal Networks or LCNs [11] were introduced recently as a probabilistic logic designed for representing and reasoning with imprecise knowledge. Unlike many existing probabilistic logics, LCNs have the ability to represent cycles (e.g., feedback loops) as well as allow specifying marginal and conditional probability bounds on logic formulae which may be important in many realistic usecases.

Up until now, the work on LCNs has focused exclusively on marginal inference, i.e. efficiently computing posterior lower and upper probability bounds on a query formula. However, *abductive reasoning* tasks such as explaining the evidence observed in an LCN are equally important in many real-world applications. In probabilistic graphical models, these tasks are commonly known as MAP and Marginal MAP (MMAP) inference and have received extensive attention over the past decades [12, 13]. They are typically tackled efficiently with dynamic programming (e.g., variable elimination) or heuristic search (e.g., depth-first branch and bound) based algorithms [13, 14, 15, 16].

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

Contribution. In this paper, we consider solving MAP and Marginal MAP inference queries in LCNs. Unlike in graphical models, an LCN encodes a set of probability distributions over its interpretations. Therefore, a complete or a partial explanation of the evidence which represents a complete or a partial truth assignment to the LCN's propositions may correspond to more than one distribution. Our work builds on very recent work on Marginal MAP inference for credal networks, a class of probabilistic graphical models that allow reasoning with imprecise probabilities [17]. We formally introduce the MAP and Marginal MAP tasks for LCNs as finding a complete or a partial truth assignment to the LCN's propositions with maximum *lower* (respectively, *upper*) probability, given some evidence in the LCN. We show how to evaluate such MAP assignments using exact marginal inference for LCNs and, subsequently, propose several search schemes based on depth-first search, limited discrepancy search and simulated annealing to solve these tasks in practice. We then extend a recent message-passing scheme for approximate marginal inference in LCNs [18] to handle effectively the MAP and MMAP inference tasks in LCNs as well as adapt the limited discrepancy search and simulated annealing methods to use an approximate evaluation of the MAP assignments during search. We experiment and evaluate our proposed exact and approximate algorithms on several classes of LCNs including random as well as more realistic LCN instances. Our results show that the search methods based on exact evaluation of the MAP assignments are limited to solving small size problems in practice, while the approximate message-passing scheme and, to some extent, the approximate search-based methods can scale to much larger problem instances. This is important because it allows us to tackle practical problems involving hundreds and possibly many thousands of propositions. The supplementary material includes additional details and experiments.

2 Background

We provide next a brief overview of basic concepts about LCNs and marginal inference in these models. Throughout the paper we will use the following notations. Logical propositions are denoted by uppercase letters (e.g., $A, B, C, ...$) while for sets of propositions we use boldfaced uppercase letters (e.g., $\mathbf{A}, \mathbf{B}, \mathbf{C}, ...$). Truth assignments to propositions (i.e., *literals*) are denoted by either lowercase or uppercase letters, namely we use a or A to indicate that proposition A holds true, and $\neg a$ or $\neg A$ if A is false. Sets of literals are denoted by boldfaced lowercase letters (e.g., $\mathbf{a}, \mathbf{b}, \mathbf{c}, ...$).

2.1 Logical Credal Networks

A Logical Credal Network (LCN) [11] is defined by a tuple $\mathcal{L} = \langle \mathbf{A}, \mathcal{C} \rangle$, where $\mathbf{A} = \{A_1, \dots, A_n\}$ is a set of propositions (or atoms), and $\mathcal{C}$ is a set of probability labeled sentences (or constraints) having the following two forms:

$$\alpha \leq P(\phi) \leq \beta \tag{1}$$

$$\alpha \leq P(\phi|\varphi) \leq \beta \tag{2}$$

Here, ϕ and φ are arbitrary propositional logic formulae[1] involving propositions in $\mathcal{A}$ and logical connectives such as negation, disjunction and conjunction, and $0 \leq \alpha \leq \beta \leq 1$ are lower and upper probability bounds, respectively.

An LCN is associated with *primal graph* which is a directed graph G containing *formula nodes* and *proposition nodes*, as well as directed edges from each proposition node in a formula ϕ to the formula node representing ϕ (for type 1 sentences), and directed edges from each of the proposition nodes in φ to φ, a directed edge from φ to ϕ, and bi-directed edges from ϕ to the proposition nodes in ϕ, respectively (for type 2 sentences) [11]. A *parent* of a proposition A in G is a proposition B such that there is a directed path in G from B to A in which all intermediate nodes are formulae. A *descendant* of a proposition A in G is a proposition B such that there is a directed path in G from A to B in which no intermediate node is a parent of A [11].

An LCN is endowed with a *Local Markov Condition* (LMC) where a proposition node A is independent, given its parents, of all proposition nodes that are not A itself nor descendants of A nor parents of A [11]. Therefore, an LCN represents a set of probability distributions over all interpretations of its formulae that satisfy the constraints represented by the type (1) and (2) sentences as well as the constraints induced by the independence relations given by the local Markov condition [11].

[1]The original definition of LCNs allows for relational structures and first-order logic formulae, but their semantics is obtained by grounding on finite domains thus yieling a propositional LCN [11].

$0.05 \leq P(B) \leq 0.1$

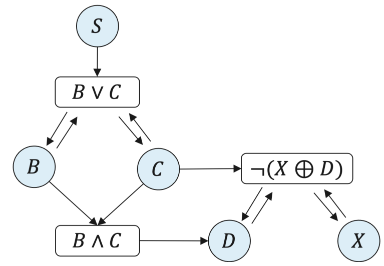
$0.3 \leq P(S) \leq 0.4$

$0.1 \leq P(B \vee C|S) \leq 0.2$

$0.6 \leq P(D|B \wedge C) \leq 0.7$

$0.7 \leq P(\neg(X \oplus D)|C) \leq 0.8$

(a) LCN sentences



(b) Primal graph

Figure 1: A simple LCN and its primal graph.

Example 1. *Figure 1 describes a simple LCN whose sentences shown in Figure 1a state that: Bronchitis (B) is more likely than Smoking (S); Smoking may cause Cancer (C) or Bronchitis; Dyspnea (D) or shortness of breadth is a common symptom for Cancer and Bronchitis; in case of Cancer we have either a positive X-Ray result (X) and Dyspnea, or a negative X-Ray and no Dyspnea. Figure 1b shows the primal graph where the formula and proposition nodes are displayed as rectangles and shaded circles, respectively.*

2.2 Marginal Inference in Logical Credal Networks

$$\sum_{i=1}^{m} p_i = 1 \tag{3}$$

$$p_i \geq 0, \forall i = 1, \dots, m \tag{4}$$

$$\alpha \leq \vec{I}_\phi \odot \vec{p} \leq \beta \tag{5}$$

$$\alpha \cdot \vec{I}_\varphi \odot \vec{p} \leq \vec{I}_{\phi \wedge \varphi} \odot \vec{p} \leq \beta \cdot \vec{I}_\varphi \odot \vec{p} \tag{6}$$

$$(\vec{I}_a \odot \vec{p}) \cdot (\vec{I}_b \odot \vec{p}) - (\vec{I}_c \odot \vec{p}) \cdot (\vec{I}_d \cdot \vec{p}) = 0 \tag{7}$$

$$\text{minimize/maximize } \vec{I}_\psi \odot \vec{p} \tag{8}$$

Given an LCN $\mathcal{L}$ with n propositions, the *marginal inference* task is to compute lower and upper bounds on the posterior probability $P(\psi)$ of a query formula ψ, which we denote by $\underline{P}(\psi)$ and $\overline{P}(\psi)$, respectively. This is achieved by solving a non-linear program given by Equations (3)–(8) and defined by a set of non-negative real-valued variables representing the probabilities of $\mathcal{L}$'s interpretations, a set of linear constraints derived from $\mathcal{L}$'s sentences, a set of non-linear constraints corresponding to the independence assumptions given by the local Markov condition, and a linear objective function encoding the query $P(\psi)$ which is minimized and maximized to yield the desired bounds. More specifically, let $\vec{p} = (p_1, \dots, p_m)$ be the vector of real-valued variables representing the probabilities of $\mathcal{L}$'s interpretations, where $m = 2^n$, and let $\vec{I}_\phi = (a_1^\phi, \dots, a_m^\phi)$ be a binary vector, called an *indicator vector*, such that a_i^ϕ is 1 if formula ϕ is true in the i-th interpretation and 0 otherwise. Since the probability of a formula ϕ is the sum of the probabilities of the interpretations in which ϕ is true, we can write $P(\phi)$ as $\vec{I}_\phi \odot \vec{p}$ where $\odot$ is the dot-product of two vectors. Therefore, Equations (3) and (4) ensure that $\vec{p}$ is a valid probability distribution, Equations (5) and (6) encode the type (1) and (2) sentences in $\mathcal{L}$ while Equation 7 encodes the conditional independencies of the form $P(X_j|\mathbf{S}_j, \mathbf{T}_j) = P(X_j|\mathbf{S}_j)$, where X_j is a proposition, $\mathbf{S}_j = \{S_{j1}, \dots, S_{jk}\}$ and $\mathbf{T}_j = \{T_{j1}, \dots, T_{jl}\}$ are X_j's parents and non-descendants in the primal graph of $\mathcal{L}$, $\vec{I}_\phi$ and $\vec{I}_{\phi\wedge\varphi}$ are the indicator vectors for formulae ϕ and $\phi \wedge \varphi$ involved in $\mathcal{L}$'s sentences, and $\vec{I}_a$, $\vec{I}_b$, $\vec{I}_c$ and $\vec{I}_d$ are the indicator vectors corresponding to the formulae $a = (x_j \wedge s_{j1} \wedge \dots \wedge s_{jk} \wedge t_{j1} \wedge \dots \wedge t_{jl})$, $b = (s_{j1} \wedge \dots \wedge s_{jk})$, $c = (x_j \wedge s_{j1} \wedge \dots \wedge s_{jk})$, and $d = (s_{j1} \wedge \dots \wedge s_{jk} \wedge t_{j1} \wedge \dots \wedge t_{jl})$, respectively (see also [11] for more details).

3 MAP and Marginal MAP Inference in LCNs

Maximum A Posteriori (MAP) and Marginal MAP (MMAP) inference are well known abductive reasoning tasks in probabilistic graphical models such as Bayesian networks and Markov networks [12, 13, 14, 15, 16]. Specifically, the MAP task calls for finding a complete assignment to all variables having maximum probability, given the evidence. Marginal MAP generalizes MAP and looks for a partial variable assignment that has maximum marginal probability, given the evidence. MAP and MMAP inference tasks appear in many real-world applications such as diagnosis, abduction and explanation and are typically tackled with dynamic programming (e.g., variable elimination) or heuristic search (e.g., depth-first branch and bound) based algorithms [13, 14, 15, 16].

In this section, we present our novel approach for solving the MAP and Marginal MAP inference tasks in Logical Credal Networks. Unlike in graphical models, a (partial) variable assignment (or interpretation) in an LCN may correspond to more than one distribution. Therefore, we begin by formally defining two MAP and MMAP inference tasks for LCNs, called *maximin MAP* (resp. *maximin MMAP*) and *maximax MAP* (resp. *maximax MMAP*). Subsequently, we develop several exact and approximation schemes for solving these tasks efficiently in practice.

3.1 The MAP and Marginal MAP Tasks in LCNs

Let $\mathcal{L} = \langle \mathbf{A}, \mathcal{C} \rangle$ be an LCN with n propositions and let $\mathbf{E} = \{E_1, \dots, E_k\} \subseteq \mathbf{A}$ be a subset of k propositions, called *evidence*, for which the truth values $\mathbf{e} = \{e_1, \dots, e_k\}$ are known. Let $\mathbf{Y} = \{Y_1, \dots, Y_m\} \subseteq \mathbf{A} \setminus \mathbf{E}$ be a subset of m propositions called *MAP propositions*. A truth assignment to $\mathbf{Y}$ is is called a *MAP assignment* and is denoted by $\mathbf{y} = \{y_1, \dots, y_m\}$, respectively. Clearly, if $\mathbf{Y} = \mathbf{A} \setminus \mathbf{E}$ (i.e., $m = n - k$) then we have a MAP task, otherwise we have a MMAP task (i.e., $m < n - k$). The *maximin* and *maximax* MAP/MMAP tasks are defined as follows:

Definition 1 (maximin)**.** *Given an LCN $\mathcal{L}$ with n propositions, evidence $\mathbf{e}$, and MAP propositions $\mathbf{Y}$, the* maximin MAP *(or* maximin MMAP *if $m < n - k$) task is finding a truth assignment $\mathbf{y}^*$ to $\mathbf{Y}$ having maximum* lower probability*, given evidence $\mathbf{e}$, namely:*

$$\mathbf{y}^* = \underset{\mathbf{y} \in \Omega(\mathbf{Y})}{\operatorname{argmax}} \underline{P}(\psi_{\mathbf{y} \wedge \mathbf{e}}) \tag{9}$$

where $\Omega(\mathbf{Y})$ is the set of all truth assignments to the MAP propositions, and $\psi_{\mathbf{y} \wedge \mathbf{e}} = y_1 \wedge \dots \wedge y_m \wedge e_1 \wedge \dots \wedge e_k$ is the conjunction of the literals in $\mathbf{y}$ and $\mathbf{e}$, respectively.

Definition 2 (maximax)**.** *Given an LCN $\mathcal{L}$ with n propositions, evidence $\mathbf{e}$, and MAP propositions $\mathbf{Y}$, the* maximax MAP *(or* maximax MMAP *if $m < n - k$) task is finding a truth assignment $\mathbf{y}^*$ to $\mathbf{Y}$ having maximum* upper probability*, given evidence $\mathbf{e}$, namely:*

$$\mathbf{y}^* = \underset{\mathbf{y} \in \Omega(\mathbf{Y})}{\operatorname{argmax}} \overline{P}(\psi_{\mathbf{y} \wedge \mathbf{e}}) \tag{10}$$

where $\Omega(\mathbf{Y})$ is the set of all truth assignments to the MAP propositions, and $\psi_{\mathbf{y} \wedge \mathbf{e}} = y_1 \wedge \dots \wedge y_m \wedge e_1 \wedge \dots \wedge e_k$ is the conjunction of the literals in $\mathbf{y}$ and $\mathbf{e}$, respectively.

3.2 Search Algorithms Using Exact MAP Assignment Evaluations

We present next three search-based schemes for solving the MAP and MMAP tasks in LCNs. These methods employ different search strategies for exploring the search space defined by the MAP propositions while evaluating exactly each complete or partial MAP assignment.

Exact Evaluation of a MAP Assignment. Clearly, computing the lower and upper probabilities $\underline{P}(\psi_{\mathbf{y} \wedge \mathbf{e}})$ and $\overline{P}(\psi_{\mathbf{y} \wedge \mathbf{e}})$ of a MAP assignment $\mathbf{y}$ given evidence $\mathbf{e}$ can be done easily by minimizing and, respectively maximizing the non-linear program defined by Equations (3)–(8), where the query formula is the conjunction of positive or negative literals in $\mathbf{y}$ and $\mathbf{e}$, namely $\psi_{\mathbf{y} \wedge \mathbf{e}} = y_1 \wedge \dots \wedge y_m \wedge e_1 \wedge \dots \wedge e_k$. Therefore, evaluating a MAP assignment in case of both MAP and Marginal MAP inference in LCNs is quite difficult as it involves solving a maginal inference problem for LCNs which is know to be NP-hard [11]. This is in contrast with graphical models where, at least for MAP inference, the evaluation of a MAP assignment is linear in the number of variables [13].

Example 2. *For illustration, consider the LCN example from Figure 1 and assume that we have evidence $\mathbf{e} = \{x, \neg s\}$, namely a patient has a positive X-Ray result ($X = x$) and is not smoking ($S = \neg s$). The MAP propositions in this case are $\mathbf{Y} = \{B, C, D\}$ and the MAP assignment $\mathbf{y} = (b, \neg c, \neg d)$ corresponds to the query formula $\psi_{\mathbf{y} \wedge \mathbf{e}} = b \wedge \neg c \wedge \neg d \wedge x \wedge \neg s$. The lower and upper probabilities $\underline{P}(\psi_{\mathbf{y} \wedge \mathbf{e}})$ and $\overline{P}(\psi_{\mathbf{y} \wedge \mathbf{e}})$ of the MAP assignment are 9.9e-09 and 0.1, respectively.*

Algorithm 1 Depth-First Search for MAP and Marginal MAP Inference in LCNs

procedure DFS($\mathcal{L} = \langle \mathbf{A}, \mathcal{C} \rangle$, $\mathbf{E} = \mathbf{e}$, $\mathbf{Y}$)
  initialize $\mathbf{y}^* \leftarrow \emptyset$, $best \leftarrow -\infty$
  SEARCH($\emptyset$, $\mathbf{Y}$)
  return $\mathbf{y}^*$
procedure SEARCH($\mathbf{y}$, $\mathbf{Y}$)
  if $size(\mathbf{y}) == size(\mathbf{Y})$ **then**
    if `maximin` **then**
      $score(\mathbf{y}) \leftarrow \underline{P}(\psi_{\mathbf{y} \wedge \mathbf{e}})$
    else
      $score(\mathbf{y}) \leftarrow \overline{P}(\psi_{\mathbf{y} \wedge \mathbf{e}})$
    if $score(\mathbf{y}) > best$ **then**
      $\mathbf{y}^* \leftarrow \mathbf{y}$, $best \leftarrow score(\mathbf{y})$
  else
    select unassigned proposition $Y_i \in \mathbf{Y}$
    for all values $y \in \{y_i, \neg y_i\}$ **do**
      $\mathbf{y} \leftarrow \mathbf{y} \cup \{Y_i = y\}$
      SEARCH($\mathbf{y}$, $\mathbf{Y}$)

Algorithm 2 Limited Discrepancy Search for MAP and Marginal MAP Inference in LCNs

1: **procedure** LDS($\mathcal{L} = \langle \mathbf{A}, \mathcal{C} \rangle$, $\mathbf{E} = \mathbf{e}$, $\mathbf{Y}$, δ)
2:   initialize $\mathbf{y}_0$ randomly and let $\mathbf{y}^* \leftarrow \mathbf{y}_0$
3:   $best \leftarrow score(\mathbf{y}^*)$
4:   **for all** $\theta = 1 \dots \delta$ **do**
5:     SEARCH($\mathbf{y}^*$, $\mathbf{Y}$, θ, 1)
6:   **return** $\mathbf{y}^*$, $best$
7: **procedure** SEARCH($\mathbf{y}$, $\mathbf{Y}$, θ, i)
8:   **if** $\theta == 0$ or $i > |\mathbf{Y}|$ **then**
9:     **if** `maximin` **then**
10:       $score(\mathbf{y}) \leftarrow \underline{P}(\psi_{\mathbf{y} \wedge \mathbf{e}})$
11:     **else**
12:       $score(\mathbf{y}) \leftarrow \overline{P}(\psi_{\mathbf{y} \wedge \mathbf{e}})$
13:     **if** $score(\mathbf{y}) > best$ **then**
14:       $\mathbf{y}^* \leftarrow \mathbf{y}$, $best \leftarrow score(\mathbf{y})$
15:   **else**
16:     **for all** values $y \in \{y_i, \neg y_i\}$ **do**
17:       **if** $\mathbf{y}[i] == y$ **then**
18:         $\mathbf{z} \leftarrow$ SEARCH($\mathbf{y}$, $\mathbf{Y}$, $i + 1$, θ)
19:       **else**
20:         $\mathbf{y}' \leftarrow \mathbf{y}$; $\mathbf{y}'[i] \leftarrow y$
21:         $\mathbf{z} \leftarrow$ SEARCH($\mathbf{y}'$, $\mathbf{Y}$, $i + 1$, $\theta - 1$)
22:       **return** $\mathbf{z}$

Depth-First Search. Our first approach for solving the MAP and MMAP tasks, called DFS, is described by Algorithm 1. It takes as input an LCN $\mathcal{L} = \langle \mathbf{A}, \mathcal{C} \rangle$, evidence $\mathbf{E} = \mathbf{e}$ and a set of MAP propositions $\mathbf{Y} \subseteq \mathbf{A} \setminus \mathbf{E}$ and outputs the optimal MAP assignment $\mathbf{y}^*$. The method conducts a *depth-first search* over the space of partial assignments to the MAP propositions, and, for each complete MAP assignment $\mathbf{y}$ computes its score as the exact lower probability $\underline{P}(\psi_{\mathbf{y} \wedge \mathbf{e}})$ for maximin tasks, and respectively, the upper probability $\overline{P}(\psi_{\mathbf{y} \wedge \mathbf{e}})$ for maximax tasks, given the evidence $\mathbf{e}$. This way, the optimal solution $\mathbf{y}^*$ corresponds to the MAP assignment with the highest score.

Theorem 1 (complexity)**.** *Given an LCN $\mathcal{L} = \langle \mathbf{A}, \mathcal{C} \rangle$ with n propositions, evidence $\mathbf{E} = \mathbf{e}$ and MAP propositions $\mathbf{Y} \subseteq \mathbf{A} \setminus \mathbf{E}$, algorithm* DFS *is sound and complete. The time and space complexity of the algorithm is $O(2^{m+2^n})$ and $O(2^n)$, respectively, where m is the number of MAP propositions.*

Example 3. *Consider again the LCN from Figure 1 with evidence $\mathbf{e} = \{x, \neg s\}$. In this case, the exact maximin MAP assignment found by algorithm* DFS *is $\mathbf{y}^* = \{\neg b, c, d\}$ with value 9.99e-09, while the exact maximax MAP assignment is $\mathbf{y}^* = \{\neg b, \neg c, d\}$ with value 0.7, respectively.*

Limited Discrepancy Search. Our second approach for MAP and MMAP inference in LCNs uses Limited Discrepancy Search (LDS) [19, 20] to explore the search space and is described by Algorithm 2. Specifically, LDS is a depth-first search strategy that searches for new solutions by iteratively increasing the number of *discrepancy* values, where a discrepancy value indicates the maximum number of allowed variable-value assignment changes to an initial solution [19]. Function SEARCH (lines 7–22) performs the actual exploration of the search space limited by discrepancy θ. If the selected truth value $y \in \{y_i, \neg y_i\}$ is different from the one corresponding to proposition $Y_i \in \mathbf{Y}$ at position i in the assignment $\mathbf{y}$, θ is decremented to reduce the number of changes allowed to the remaining MAP propositions. Otherwise, the truth value for proposition Y_i remains unchanged and the θ value is preserved. As before, complete MAP assignments are evaluated exactly (lines 9–12) and the best solution found so far is maintained (lines 13-14).

Algorithm 3 Simulated Annealing for MAP and Marginal MAP Inference in LCNs

procedure SA($\mathcal{L} = \langle \mathbf{A}, \mathcal{C} \rangle$, $\mathbf{E} = \mathbf{e}$, $\mathbf{Y}$)
  initialize $\mathbf{y}_0$ randomly and let $\mathbf{y}^* \leftarrow \mathbf{y}_0$
  $best \leftarrow score(\mathbf{y}^*)$
  for all iterations $i = 1 \dots N$ **do**
    set $\mathbf{y} \leftarrow \mathbf{y}^*$, $T \leftarrow T_{init}$
    for all flips $j = 1 \dots M$ **do**
      let $\mathcal{N}$ be $\mathbf{y}$'s neighbors
      select random neighbor $\mathbf{y}' \in \mathcal{N}$
      $\Delta \leftarrow \log score(\mathbf{y}') - \log score(\mathbf{y})$
      if $\Delta > 0$ **then** $\mathbf{y} \leftarrow \mathbf{y}'$
      else
        sample randomly $p \in (0, 1)$
        if $p < e^{\frac{\Delta}{T}}$ **then** $\mathbf{y} \leftarrow \mathbf{y}'$
      if $score(\mathbf{y}) > best$ **then**
        $\mathbf{y}^* \leftarrow \mathbf{y}$, $best \leftarrow score(\mathbf{y})$
      $T \leftarrow T * \sigma$
  return $\mathbf{y}^*$

Algorithm 4 Approximate MAP and Marginal MAP Inference in LCNs

procedure AMAP($\mathcal{L} = \langle \mathbf{A}, \mathcal{C} \rangle$, $\mathbf{E} = \mathbf{e}$, $\mathbf{Y}$)
  Create factor graph $\mathcal{F}$ of $\mathcal{L}$
  Apply the ARIEL scheme from [18] on $\mathcal{F}$
  for all MAP propositions $Y \in \mathbf{Y}$ **do**
    if `maximin` **then**
      $\underline{P}(y) = \max_{f \in N(Y)} l_{f \to Y}$
      $\underline{P}(\neg y) = 1 - \underline{P}(y)$
      if $\underline{P}(y) > \underline{P}(\neg y)$ **then** $\mathbf{y}^* \leftarrow \mathbf{y}^* \cup \{y\}$
      else $\mathbf{y}^* \leftarrow \mathbf{y}^* \cup \{\neg y\}$
    else
      $\overline{P}(y) = \min_{f \in N(Y)} u_{f \to Y}$
      $\overline{P}(\neg y) = 1 - \overline{P}(y)$
      if $\overline{P}(y) > \overline{P}(\neg y)$ **then** $\mathbf{y}^* \leftarrow \mathbf{y}^* \cup \{y\}$
      else $\mathbf{y}^* \leftarrow \mathbf{y}^* \cup \{\neg y\}$
  return $\mathbf{y}^*$

Theorem 2 (complexity). *Given an LCN $\mathcal{L} = \langle \mathbf{A}, \mathcal{C} \rangle$ with n propositions, evidence $\mathbf{E} = \mathbf{e}$ and MAP propositions $\mathbf{Y} \subseteq \mathbf{A} \setminus \mathbf{E}$, algorithm* LDS *is sound and complete. The time and space complexity of the algorithm is $O(2^{m+2^n})$ and $O(2^n)$, respectively, where m is the number of MAP propositions.*

Simulated Annealing. The third approach for solving MAP and MMAP tasks in LCNs is described by Algorithm 3 and employs a form of stochastic local search known as Simulated Annealing (SA) [21] to explore the search space defined by the MAP propositions. The algorithm starts from an initial guess $\mathbf{y}$ as a truth assignment to the MAP propositions $\mathbf{Y}$, and iteratively tries to improve it by moving to a better neighbor $\mathbf{y}'$ that has a higher score. A *neighbor* $\mathbf{y}'$ of $\mathbf{y}$ is defined as a new assignment $\mathbf{y}'$ which results from changing the truth value of a single proposition Y in $\mathbf{Y}$. At each step, the transition from the current state $\mathbf{y}$ to a neighboring state $\mathbf{y}'$ is decided probabilistically using an acceptance probability function $P(\mathbf{y}', \mathbf{y}, T)$ that depends on the scores of the two states as well as a global time-varying parameter T called *temperature* which is decreased using a cooling schedule $\sigma < 1$ [21]. We chose $P(\mathbf{y}', \mathbf{y}, T) = e^{\frac{\Delta}{T}}$, where $\Delta = \log score(\mathbf{y}') - \log score(\mathbf{y})$.

Theorem 3 (complexity). *Given an LCN $\mathcal{L} = \langle \mathbf{A}, \mathcal{C} \rangle$ with n propositions, evidence $\mathbf{E} = \mathbf{e}$ and MAP propositions $\mathbf{Y} \subseteq \mathbf{A} \setminus \mathbf{E}$, the time and space complexity of algorithm* SA *is $O(N \cdot M \cdot 2^{2^n})$ and $O(2^n)$, respectively, where N is the number of iterations and M is the number of flips per iterations.*

3.3 Approximate MAP and Marginal MAP Inference

The main bottleneck in the proposed search algorithms is the exact evaluation of the MAP assignments which is computationally very expensive [11]. This limits the applicability of these methods to relatively small LCNs. Therefore, in order to be able to tackle larger LCNs, we extend a recent message-passing approximation scheme for marginal inference in LCNs [18] to solve the MAP and MMAP tasks in LCNs. Subsequently, we also adapt the limited discrepancy search and simulated annealing methods to use an approximate evaluation of the MAP assignments during search.

Algorithm 4 describes our message-passing based approximation scheme for MAP and MMAP inference in LCNs which we denote hereafter by AMAP. We build upon a recent scheme for approximate marginal inference in LCNs, called ARIEL [18], which propagates messages along the edges of a *factor graph* associated with the input LCN until convergence. The factor graph $\mathcal{F}$ of an

LCN $\mathcal{L}$ is a bi-partite graph the connects *proposition nodes* labeled by the propositions in $\mathcal{L}$ with *factor nodes* associated with sentences that involve the same set of propositions [18]. The messages propagated between the nodes of $\mathcal{F}$ are intervals representing lower and upper bounds on the marginal probabilities of $\mathcal{L}$'s propositions and are computed as follows: the message sent from a proposition to a factor node tightens these bounds based on the incoming messages from the factor nodes connected to it; the message sent from a factor to a proposition node computes new bounds by solving a local non-linear program defined by the factor's sentences and the constraints encoding the assumption that the factor's propositions are independent of each other and the marginal probabilities of the factor's propositions are within the bounds given by the incoming proposition-to-factor messages (see also [18] for more details). Upon convergence, the maximin MAP assignment $\mathbf{y}^*$ can be obtained as follows: for each MAP proposition $Y \in \mathbf{Y}$ we compute the tightest lower probability bound $\underline{P}(y)$ by maximizing the lower bound of all incoming factor-to-proposition messages to Y, and, subsequently, select y as the most likely value assignment to Y if $\underline{P}(y) > \underline{P}(\neg y)$ and $\neg y$ otherwise (for the maximax tasks we use the upper probability bounds $\overline{P}(y)$ and $\overline{P}(\neg y)$, respectively).

Theorem 4 (complexity)**.** *Given an LCN $\mathcal{L} = \langle \mathbf{A}, \mathcal{C} \rangle$ with n propositions, evidence $\mathbf{E} = \mathbf{e}$ and MAP propositions $\mathbf{Y} \subseteq \mathbf{A} \setminus \mathbf{E}$, the time and space complexity of algorithm* AMAP *is $O(N \cdot M \cdot 2^{2^r})$ and $O(2^r)$, where N is the number of iterations, M bounds the number of factor-to-node messages per iteration and r bounds the number of propositions in the factor nodes, respectively.*

3.4 Search Algorithms Based on Approximate MAP Evaluations

The main assumption behind algorithm AMAP is that all MAP propositions are independent of each other and therefore the solution $\mathbf{y}^*$ returned by AMAP is likely to correspond to a local maxima. One way to escape such a local optima and obtain a potentially better solution is to employ a search scheme based on either limited discrepancy search or simulated annealing that continues the exploration of the search space starting from $\mathbf{y}^*$. However, in order to scale to larger LCNs, we would like the search schemes to rely on an approximate rather than an exact evaluation of the MAP assignments.

Approximate Evaluation of a MAP Assignment. Estimating the lower and upper probabilities of a MAP assignment $\mathbf{y}$ can be done by approximate marginal inference on an *augmented* LCN as follows. Let $\mathcal{L} = \langle \mathbf{A}, \mathcal{C} \rangle$ be the input LCN and let $\mathbf{y} = (y_1, \dots, y_m)$ be a MAP assignment to propositions $\mathbf{Y} = \{Y_1, \dots, Y_m\}$ (for simplicity, we include the evidence $\mathbf{e}$ in $\mathbf{y}$). The *augmented* LCN $\mathcal{L}' = \langle \mathbf{A}', \mathcal{C}' \rangle$ is constructed by adding a set of auxiliary propositions $\mathbf{W} = \{W_1, \dots, W_m\}$, one for each MAP proposition, and additional constraints of the following two forms: $P(W_1|Y_1)$ and $P(W_j|W_{j-1} \wedge Y_j)$, for all $2 \leq j \leq m$, such that $P(w_1|y_1) = 1$, $P(w_1|\neg y_1) = 0$, $P(w_j|w_{j-1} \wedge y_j) = 1$, $P(w_j|w_{j-1} \wedge \neg y_j) = 0$, $P(w_j|\neg w_{j-1} \wedge y_j) = 0$ and $P(w_j|\neg w_{j-1} \wedge \neg y_j) = 0$, respectively. Then, we can estimate $\underline{P}(\psi_{\mathbf{y}})$ and $\overline{P}(\psi_{\mathbf{y}})$, where $\psi_{\mathbf{y}} = y_1 \wedge \cdots \wedge y_m$, by computing the posterior marginals $\underline{P}(w_m)$ and $\overline{P}(w_m)$ in the augmented LCN $\mathcal{L}'$ using the method from [18].

Limited Discrepancy Search and Simulated Annealing. Our approximate LDS and SA based algorithms denoted by ALDS and ASA can be obtained from Algorithms 2 and 3 by replacing the $score(\mathbf{y})$ function with the approximate MAP evaluation scheme described above. These algorithms can start the search either from a random MAP assignment or from the solution found by algorithm AMAP. Finally, the time complexity of algorithms ALDS and ASA can be bounded by $O(2^{m+2^r})$ and $O(N \cdot M \cdot 2^{2^r})$, respectively, where m is the number of MAP propositions, N is the number of iterations used by ASA, M is the maximum number of flips per iteration, and r bounds the number of propositions in the factor nodes of the factor graph associated with the input LCN [18].

4 Experiments

In this section, we empirically evaluate the proposed exact and approximate schemes for MAP and MMAP inference in LCNs. All competing algorithms were implemented[2] in Python 3.10 and used the `ipopt` 3.14 solver [22] with default settings to handle the non-linear constraint programs. We ran all experiments on a 3.0GHz Intel Core processor with 128GB of RAM.

[2]The open-source implementation of LCNs is available at: `https://github.com/IBM/LCN`

Table 1: Results for MAP tasks obtained on small/large scale polytree, dag, and random LCNs. Average CPU time in seconds and number of problem instances solved. Time limit is 2 hours.

size	exact MAP eval				approx MAP eval	
n	DFS	LDS(3)	SA	AMAP	ALDS(3)	ASA
			polytree			
5	15.30 (10)	26.07 (10)	20.18 (10)	2.87 (10)	174.17 (10)	188.27 (10)
8	3246.28 (4)	3072.18 (4)	1199.51 (10)	8.05 (10)	1054.53 (10)	518.18 (10)
10	-	-	-	11.81 (10)	2273.16 (10)	813.30 (10)
30	-	-	-	31.55 (10)	-	3091.74 (10)
50	-	-	-	52.30 (10)	-	5324.71 (10)
70	-	-	-	79.28 (10)	-	7279.56 (10)
			dag			
5	21.09 (10)	15.66 (10)	24.04 (10)	5.54 (10)	163.02 (10)	156.34 (10)
8	1633.38 (8)	1958.16 (9)	633.77 (10)	13.05 (10)	1339.71 (10)	571.55 (10)
10	-	-	-	15.55 (10)	2903.05 (10)	944.17 (10)
30	-	-	-	49.94 (10)	-	3593.71 (10)
50	-	-	-	89.13 (10)	-	5639.90 (10)
70	-	-	-	132.34 (10)	-	6093.28 (10)
			random			
5	19.51 (10)	17.56 (10)	20.37 (10)	5.26 (10)	152.99 (10)	143.60 (10)
8	3152.57 (1)	3209.54 (5)	1226.88 (10)	10.29 (10)	954.46 (10)	444.17 (10)
10	-	-	-	12.21 (10)	2150.27 (10)	717.75 (10)
30	-	-	-	40.54 (10)	-	3335.14 (10)
50	-	-	-	76.83 (10)	-	5276.93 (10)
70	-	-	-	105.70 (10)	-	6059.57 (7)

Random LCNs. We generated three classes of random LCNs with n propositions $\{X_1, \dots X_n\}$ and sentences of the following types: (a) $l \leq P(x_i) \leq u$, (b) $l \leq P(x_i|x_j) \leq u$, $i \neq j$ and (c) $l \leq P(x_i|X_j \wedge X_k) \leq u$, $i \neq j \neq k$, such that the corresponding primal graph is a polytree, a dag or a random graph. The type (c) sentences were generated for all truth values of propositions X_j and X_k, namely $P(x_i|x_j)$, $P(x_i|\neg x_j)$, $P(x_i|x_j \wedge x_k)$, $P(x_i|x_j \wedge \neg x_k)$, $P(x_i|\neg x_j \wedge x_k)$ and $P(x_i|\neg x_j \wedge \neg x_k)$, respectively. The probability bounds l and u were selected uniformly at random between 0 and 1 such that $u - l \leq 0.6$, and we ensured that all instances with $n \leq 10$ were consistent.

Table 1 summarizes the results obtained for maximax MAP queries on the random LCNs. For each problem class we consider both smaller ($5 \leq n \leq 10$) and larger ($30 \leq n \leq 70$) scale instances, respectively. We report the average CPU time in seconds and number of problem instance solved (out of 10) for each problem size. A '-' indicates that the respective algorithm exceeded the 2 hour time limit. The maximum discrepancy value use by algorithms LDS and ALDS was set to $\delta = 3$, while algorithms SA and ASA used up to 30 flips over a single iteration. We can see that the algorithms using exact MAP assignment evaluations (i.e., DFS, LDS and SA) are limited to small scale problem instances with up to 8 propositions and they run out of time on the larger instances. This is caused by the prohibitively large computational overhead associated with the exact evaluation of the MAP assignments during search. In contrast, the approximate search algorithms ALDS and specially ASA can scale to much larger problem instances due to the less expensive approximate MAP assignment evaluations. AMAP is the best performing algorithm in terms of running time and number of problems solved for all reported problem sizes. However, since the solution found by AMAP is only a local maxima, in Figure 2 we report on the solution quality found by algorithms AMAP, ALDS and ASA on LCN instances of size 10. Specifically, we show the number of wins as the number of times (out of 10) each algorithm found the best solution. In this case, algorithms ALDS and ASA were initialized with the MAP assignment

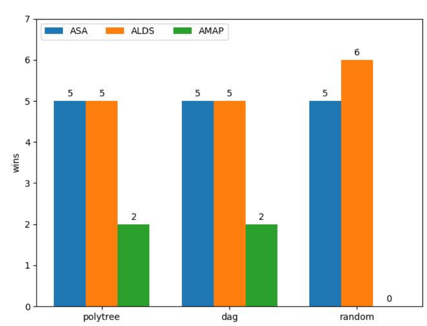


Figure 2: Wins for LCNs with $n = 10$.

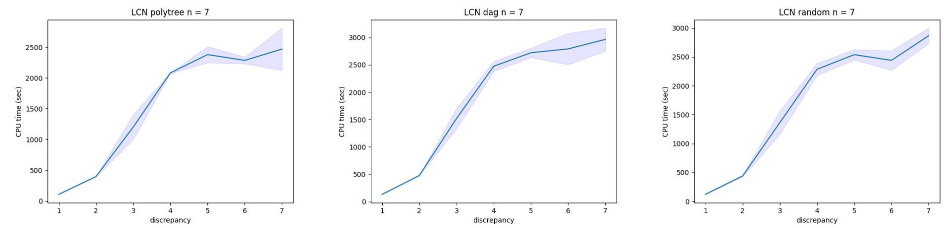


Figure 3: Average CPU time in seconds and standard deviation vs discrepancy δ for ALDS(δ).

Table 2: Results for MMAP tasks on realistic LCNs. CPU time in seconds. Time limit is 2 hours.

LCN	exact MAP eval				approx MAP eval	
	DFS	LDS(3)	SA	AMAP	ALDS(3)	ASA
Toy	2.20	3.18	1.85	0.85	134.83	141.17
Earth	9.19	7.67	2.75	1.28	150.99	162.35
Cancer	16.34	14.09	8.52	2.64	157.92	159.66
Asia	811.82	800.18	312.10	4.07	187.44	201.76
Credit	-	6719.30	2976.55	5.09	204.77	222.52
Engine	4786.12	4502.34	2033.77	6.57	212.61	235.70
Suicide	-	-	-	5.99	220.31	203.68
Tank	-	-	-	8.04	263.65	281.73
Alarm	-	-	-	4.28	216.19	186.67
Hepatitis	-	-	-	8.22	260.38	250.45

found by AMAP. We can see that almost always the search-based approaches ALDS and ASA are able to find better solutions than AMAP. This is important in practice, particularly on larger scale problems where we can use AMAP to find a MAP solution quickly, and subsequently refine that solution using a search-based algorithm like ALDS or ASA if the time budget allows it. Finally, in Figure 3 we show the impact of the maximum discrepancy value δ on the running time of algorithm ALDS(δ). It is easy to see that as the discrepancy value δ increases, the search space explored by ALDS(δ) becomes larger, and therefore its corresponding running time increases as well.

Realistic LCNs. We experimented with a set of more realistic LCNs which were first introduced in [18]. These LCNs were derived from real-world Bayesian networks [23] and contain up to 10 propositions as well as up to 24 sentences of the form $l \leq P(x_i) \leq u$ and $l \leq P(x_i|\pi_i) \leq u$, respectively, where x_i is the positive literal of proposition X_i and $\pi_i = y_{i1} \wedge \dots \wedge y_{ik}$ is the conjunction of the positive or negative literals corresponding to a particular configuration of the parents $\{Y_{i1}, \dots Y_{ik}\}$ of X_i in the Bayesian network. The specification of these LCNs is included in the supplementary material. Table 2 reports the results obtained on 10 LCN instances for the `maximax` MMAP task with 4 MAP propositions selected randomly. As before, algorithms DFS, LDS(3) and SA which rely on exact evaluations of the MAP assignments during search can only solve the smallest problem instances within the 2 hour time limit. In contrast, algorithms ALDS(3) and ASA solve all problem instances due to a much reduced overhead associated with the approximate MAP assignment evaluations. In this case, the search spaces explored by ALDS(3) and ASA are approximately the same in size and therefore the corresponding running times are comparable. AMAP is the fastest algorithm in this case as well.

Application to Factuality in Large Language Models. We consider an application of MMAP inference in LCNs to assess the factuality of the output A generated by a large language model (LLM) in response to a user query Q with respect to an external source of knowledge C that may contain contradicting factual information (e.g., Wikipedia) [24]. The goal is to compute a *factuality score* for response A, denoted by $f_C(A)$, in the context of the information from C. In the following, we assume that A can be decomposed into a set of n *atomic facts* (or just *atoms*) $A = \{A_1, \dots, A_n\}$ (e.g., one way to do that is to split A into sentences) and, for each atom A_i, up to k relevant passages $\{C_{i1}, \dots, C_{ik}\}$ called *contexts* can be retrieved from C. A natural language inference (NLI) classifier such as SBERT [25] can be used to infer the *entailment*, *contradiction* and *neutrality* relationships between the texts corresponding to the atoms and contexts together

Table 3: Results for factuality LCNs. Average CPU time in seconds and number of problem instances solved. Time limit is 2 hours.

size	exact MAP eval				approx MAP eval	
$n, k = 2$	DFS	LDS(2)	SA	AMAP	ALDS(2)	ASA
2	56.95 (10)	57.37 (10)	60.09 (10)	0.31 (10)	5.25 (10)	4.13 (10)
4	-	-	-	0.98 (10)	80.07 (10)	54.15(10)
6	-	-	-	1.97 (10)	453.88 (10)	219.57 (10)
10	-	-	-	7.33 (10)	2713.90 (10)	928.28 (10)
20	-	-	-	28.42 (10)	-	3809.23 (10)
50	-	-	-	379.18 (10)	-	-
100	-	-	-	1807.10 (10)	-	-

with their corresponding probabilities (or scores). Specifically, we consider relationships between an atom and a context $r(A_i, C_{ij})$, and between two contexts $r(C_{ij}, C_{pq})$, respectively, where $r \in \{\text{entailment}, \text{contradiction}\}$. We define an LCN $\mathcal{L}$ containing $n + n \times k$ propositions for each of the atoms and contexts, and two types of sentences corresponding to the entailment and contradiction relationships as follows: $l \leq P(Y|X) \leq u$ if X entails Y, and $l \leq P(\neg Y|X) \leq u$ if X contradicts Y, where X and Y are the propositions corresponding to a context and an atom, or to two different contexts, respectively. The lower and upper bounds l and u can be calculated easily from the probabilities obtained by running multiple NLI classifiers. Finally, the factuality score $f_C(A)$ is the proportion of true atoms in the MAP assignment obtained by solving a `maximax` MMAP task over $\mathcal{L}$ where the MAP propositions are those corresponding to A's atoms.

Table 3 displays the results obtained on randomly generated factuality LCNs. More specifically, for each reported problem size $n \in \{2, 4, 6, 10, 20, 50, 100\}$, we generated 10 random instances with n atoms and $k = 2$ contexts per atom such that 10% of all possible pairwise relationships between atoms and contexts were selected to be either *entailment* or *contradiction* with probability 0.5 while the remaining relationships were labeled as *neutral* and thus ignored. The lower and upper probability bounds l and u in the corresponding LCN sentences were also generated randomly between 0 and 1 such that $u - l \leq 0.6$. In this case, the maximum discrepancy value was set to 2 and simulated annealing was allowed a single iteration and 30 flips. We observe again that algorithms DFS, LDS(2) and SA can only solve the smallest instances due to large computational overhead associated with exact evaluation of the MAP assignments. In contrast, algorithms ALDS(2) and ASA which rely on less expensive approximate evaluations of the MAP assignments can scale to larger problems with up to 20 atoms. Algorithm AMAP outperforms its competitors and solves all problem instances.

In summary, our empirical evaluation showed that the exact search-based MAP/MMAP algorithms are limited to solving relatively small problem instances. In contrast, the approximate MAP/MMAP schemes based on either message-passing or search can scale to much larger LCN instances.

5 Conclusions

In this paper, we address abductive reasoning tasks such as generating MAP and Marginal MAP (MMAP) explanations in Logical Credal Networks (LCNs), a recently introduced probabilistic logic framework for reasoning with imprecise knowledge. Since an LCN encodes a set of distributions over its interpretations, a complete or partial explanation of the evidence (i.e., a MAP assignment) may correspond to more than one distribution. Therefore, we define the maximin/maximax MAP and MMAP tasks for LCNs as finding complete or partial MAP assignments that have maximum lower/upper probability given the evidence. We propose several search algorithms that combine depth-first search, limited-discrepancy search or simulated annealing with exact evaluations of the MAP assignments using marginal inference for LCNs. We also develop an approximate message-passing scheme as well as extend limited discrepancy search and simulated annealing to use an approximate evaluation of the MAP assignments during search. Our experiments with random LCNs and LCNs derived from realistic use-cases demonstrate conclusively that the search methods based on exact evaluations of the MAP assignments are limited to small size problems, while the approximation schemes can scale to much larger problems. For future work we plan to investigate more advanced depth-first branch-and-bound and best-first search techniques. However, these kinds of methods require developing novel heuristic bounding schemes to guide the search more effectively [16].

Acknowledgements

Fabio Cozman thanks CNPq (grant 305753/2022-3) and the Center for AI at Universidade de São Paulo, funded by FAPESP (grant 2019/07665-4) and IBM.

References

[1] Nils Nilsson. Probabilistic logic. *Artificial Intelligence*, 28(1):71–87, 1986.

[2] Ronald Fagin, Joseph Halpern, and Nimrod Megiddo. A logic for reasoning about probabilities. *Information and Computation*, 87(1-2):78–128, 1990.

[3] Jochen Heinsohn. Probabilistic description logics. In *Proceedings of the International Conference on Uncertainty in Artificial Intelligence*, pages 311–318, 1994.

[4] Manfred Jaeger. Probabilistic reasoning in terminological logics. In *Principles of Knowledge Representation and Reasoning*, pages 305–316. Elsevier, 1994.

[5] Kent Andersen and John Hooker. Bayesian logic. *Decision Support Systems*, 11(2):191–210, 1994.

[6] Vijay Chandru and John Hooker. *Optimization Methods for Logical Inference*. John Wiley & Sons, 1999.

[7] Michael Dürig and Thomas Studer. Probabilistic abox reasoning: Preliminary results. In *Description Logics*, pages 104–111, 2005.

[8] Matthew Richardson and Pedro Domingos. Markov logic networks. *Machine Learning*, 62(1-2):107–136, 2006.

[9] Lise Getoor and Ben Taskar. *Introduction to Statistical Relational Learning (Adaptive Computation and Machine Learning)*. MIT Press, 2007.

[10] Luc De Raedt, Paolo Frasconi, Kristian Kersting, and Stephen Muggleton. *Probabilistic Inductive Logic Programming - Theory and Applications*. Springer, 2008.

[11] Radu Marinescu, Haifeng Qian, Alexander Gray, Debarun Bhattacharjya, Francisco Barahona, Tian Gao, Ryan Riegel, and Pravinda Sahu. Logical credal networks. In *36th Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

[12] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, 1988.

[13] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[14] Radu Marinescu and Rina Dechter. AND/OR branch-and-bound search for combinatorial optimization in graphical models. *Artificial Intelligence*, 173(16-17):1457–1491, 2009.

[15] Radu Marinescu and Rina Dechter. Memory intensive AND/OR search for combinatorial optimization in graphical models. *Artificial Intelligence*, 173(16-17):1492–1524, 2009.

[16] Radu Marinescu, Junkyu Lee, Rina Dechter, and Alexander Ihler. AND/OR search for marginal MAP. *Journal or Artificial Intelligence Research (JAIR)*, 63(1):875 – 921, 2018.

[17] Radu Marinescu, Debarun Bhattacharjya, Junkyu Lee, Alexander Gray, and Fabio Cozman. Credal marginal map. In *37th Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

[18] Radu Marinescu, Haifeng Qian, Alexander Gray, Debarun Bhattacharjya, Francisco Barahona, Tian Gao, and Ryan Riegel. Approximate inference in logical credal networks. In *32nd International Joint Conference on Artificial Intelligence (IJCAI)*, 2023.

[19] William Harvey and Matthew Ginsberg. Limited discrepancy search. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 607–613, 1995.

[20] Richard Korf. Improved limited discrepancy search. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 286–291, 1996.

[21] Scott Kirkpatrick, Daniel Gelatt, and Mario Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[22] Andreas Wächter and Lorenz Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 2006.

[23] Anthony Constantinou, Yang Liu, Kiattikun Chobtham, Zhigao Guo, and Neville Kitson. The bayesys data and bayesian network repository. Technical report, Bayesian Artificial Intelligence research lab, Queen Mary University of London, London, UK, 2020.

[24] Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, 2023.

[25] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pages 3973–3983, 2019.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and precede the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS paper checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Sections 3 and 4

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: Section 3, 4 and 5. Essentially the exact inference methods proposed in this paper are limited to small size problems with up to 8 propositions/variables while the proposed approximate inference methods can scale to much larger problems with tens and even hundreds of variables.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Section 3

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 4 and supplementary material

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Section 4 and supplementary material

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4 and the supplementary material

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Section 4 and the supplementary material

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section 4

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification:

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification:

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification:

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Section 4

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Section 4

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.