Probabilistic Logic with Strong Independence

Fabio G. Cozman[1], Cassio P. de Campos[1,2], and José Carlos F. da Rocha[3]

[1] Escola Politécnica, Univ. de São Paulo, São Paulo, SP, Brazil.
[2] Univ. Mackenzie, São Paulo, SP, Brazil.
[3] Univ. Estadual de Ponta Grossa, Ponta Grossa, PR, Brazil.
fgcozman@usp.br, cassio@ime.usp.br, jrocha@uepg.br

Abstract. This papers investigates the manipulation of statements of strong independence in probabilistic logic. Inference methods based on polynomial programming are presented for strong independence, both for unconditional and conditional cases. We also consider graph-theoretic representations, where each node in a graph is associated with a Boolean variable and edges carry a Markov condition. The resulting model generalizes Bayesian networks, allowing probabilistic assessments and logical constraints to be mixed.

1 Introduction

Probabilistic logic offers a unifying language for a vast portion of human and computational discourse, as it merges logical and probabilistic sentences into a uniform scheme. However, probabilistic logic faces two difficulties: *inferential vacuousness* and *computational complexity*.

A simple example of inferential vacuousness is as follows. Suppose events A and B have no logical relation and $P(A) = p$, $P(B) = q$, with $p, q > 0$. Then the probability of $P(A \wedge B)$ is *completely* vacuous: $P(A \wedge B)$ can be in the whole interval $[0, 1]$. In this simple example it is obvious that a judgement of independence would greatly change matters: if the events A and B are considered independent, then $P(A \wedge B) = pq$. In fact, judgements of independence are used in large multivariate models, such as Bayesian networks, to create complex distributions out of few probability assessments.

The goal of this paper is to pursue a language that retains both the freedom of probabilistic logic and the power of independence relations. In short, the thesis explored in this paper is that independence relations should be part of the vocabulary of general probabilistic logic and probabilistic logic programming. The adopted language is the propositional part of Halpern's first order probabilistic logic [1, 2], plus a few predicates that are needed to deal with independence. The contribution is to propose concrete algorithms that can handle this extended vocabulary.

As for computational complexity, probabilistic logic without independence relations is already quite hard. Independence relations cannot per si reduce this complexity; however, independence relations may allow us to build compact and modular knowledge bases, hopefully suggesting simplifications and approximations that are not available in independence-free probabilistic logic. This paper explores graph-theoretic representations for large multivariate models, aiming at a language that can expose such modularity in practice.

Section 2 presents a few relevant facts about probabilistic logic. Section 3 treats the inference problem with strong independence judgements and unstructured sentences, while Section 4 defines a graph-theoretic representation for probabilistic and logical assessments and presents an inference algorithm. Section 5 concludes the paper.

2 Probabilistic Logic

In this paragraph we fix basic notation; there are detailed treatments of propositional and first order logic in the literature [3]. We use a set of *propositions* $\mathbb{A} = \{A_i\}_{i=1}^n$, taken as facts, situations, or events. An *atomic formula* consists of a proposition. A *formula* consists either of an atomic formula or of a combination of other formulas through logical connectives: negation ($\neg$), disjunction ($\vee$) and conjunction ($\wedge$). Connectives have the usual semantics defined by truth tables. A *literal* is a formula consisting either of a proposition or a negated proposition. A *truth assignment* to a set of propositions is a vector assigning either value true or value false to each proposition (and each proposition can take only one of these values). These assignments are often called *possible worlds*. For n propositions, there are 2^n truth assignments; if a formula ϕ is true for some truth assignment w, then ϕ is satisfied by w.

The *probabilistic satisfiability* problem is this: given a set of propositions $\mathbb{A}$, a set of formulas $\{\phi_i\}_{i=1}^m$ over $\mathbb{A}$, and a set of assessments p_i in the real interval $[0, 1]$, such that each p_i is associated with a formula ϕ_i, is there a probability distribution P such that $P(\phi_i) = p_i$ for all p_i [4, 5]? Here $P(\phi_i)$ is taken as the measure of the set of possible worlds where ϕ_i is true. A related problem is to determine the probability $P(\phi^*)$ for a formula ϕ^* given a set of formulas and assessments. This is often called a *probabilistic inference* for $P(\phi^*)$.

Probabilistic satisfiability has old roots and has been rediscovered a few times. Boole tried already to combine probability and logic [6]; Boole's efforts were then generalized by Hailperin [7], and later proposed in the artificial intelligence literature by Nilsson [8]. Linear programming is the main tool for satisfiability and inference, and it can handle assessments of conditional probability, inference of conditional probability, and assessments in the form of probability intervals. An excellent historical and technical review is given by Hansen and Jaumard [5].

A different trail to probabilistic satisfiability started with de Finetti, who began a comprehensive program to place logical and probabilistic statements into a single coherency-based framework [9]. Linear programming is again the main inference tool [10], but the coherency-based approach is distinguished by its ability to handle conditioning on propositions of zero probability [11] — in this paper we assume that every conditioning event has positive probability. The relationship between probabilistic satisfiability and the coherency-based framework has been explored recently [12].

A few authors have considered *stochastic independence*[4] in probabilistic satisfiability [13, 5]. No systematic implementation of proposed algorithms has been discussed in the literature. The study of independence in the context of coherency-based reasoning, where zero probabilities are dealt with explicitly, started recently with the seminal work

[4] Events C and D are stochastically independent if $P(C|D) = P(C)$ and $P(D|C) = P(D)$.

of Coletti and Scozzafava [14], and is explored in depth in the work of Vantaggi [15]. Specialized algorithms for small problems have been proposed in this latter work.

Two further notable generalizations of probabilistic satisfiability have appeared in the artificial intelligence literature: constructs from first order logic have been incorporated [1], and declarative programs have adopted techniques from probabilistic satisfiability [16, 17]. A rather elegant formulation of *probabilistic logic programming* is given by Lukasiewicz, where a rule is expressed as $(\phi|\varphi)[l, u]$, meaning "the probability of ϕ conditional on φ is in the interval $[l, u]$" [18]. We adopt this notation in this paper, specialized for propositional formulae. More general first order probabilistic logics have appeared in the work of Bacchus [19] and Halpern [1, 2]. These logics define model-based semantics for general formulae; in this paper we only use their propositional parts plus predicates that indicate independence (say an independent predicate) and that indicate graph-theoretic constructs (such as parentOf).

3 Inferences with Unstructured Sentences

Consider m assessments $(\phi_i|\varphi_i)[l_i, u_i]$ defined on n Boolean variables $X_1, \ldots, X_n$. We use x to indicate that variable X is true, and use $\overline{x}$ to indicate that variable X is false. There are 2^n possible worlds (each formula ϕ_i, φ_i is true or false in each world) and world w_k is associated with probability p_k. Denote by $V[\phi]$ the vector with 2^n elements where the kth element is 1 if ϕ is true in w_k and 0 otherwise, and denote by $\mathbf{p}$ the vector containing all p_k. Under the assumption of positivity for conditioning events mentioned previously, any conditional assessment can be written as constraints such as $(V[\phi_i \wedge \varphi_i] - l_i V[\varphi_i]) \cdot \mathbf{p} \geq 0$.

The inference problem is to compute the lower and upper probabilities for a formula ϕ^* conditional on a formula φ^*:

$$\min_{\mathbf{p}} / \max_{\mathbf{p}} \left(V[\phi^* \wedge \varphi^*] \cdot \mathbf{p}\right) / \left(V[\varphi^*] \cdot \mathbf{p}\right), \quad (1)$$

subject to $\sum_k p_k = 1$, $p_k \geq 0$ for all k, $(V[\phi_i \wedge \varphi_i] - l_i V[\varphi_i]) \cdot \mathbf{p} \geq 0$ and $(V[\phi_i \wedge \varphi_i] - u_i V[\varphi_i]) \cdot \mathbf{p} \leq 0$, for all $i = 1, \ldots, m$. The Charnes-Cooper transformation can reduce this optimization problem to the linear program $\min_{\mathbf{p}} / \max_{\mathbf{p}} (V[\phi^* \wedge \varphi^*] \cdot \mathbf{p})$, subject to $V[\varphi^*] \cdot \mathbf{p} = 1$, $p_k \geq 0$ for all k, $(V[\phi_i \wedge \varphi_i] - l_i V[\varphi_i]) \cdot \mathbf{p} \geq 0$ and $(V[\phi_i \wedge \varphi_i] - u_i V[\varphi_i]) \cdot \mathbf{p} \leq 0$, for all $i = 1, \ldots, m$. The Charnes-Cooper transformation is well-known in probabilistic logic [5] and statistics [20].

Note that probabilistic logic is essentially concerned with sets of probability measures — sets induced by linear and fractional equality and inequality constraints. To this model we will include independence judgements. There are several concepts of independence that can be used when one deals with sets of probabilities [21–23]. We adopt the most commonly used concept of *strong independence*: formulas θ and ϑ are *strongly independent* conditional on η when

$$\left(V[\theta \wedge \vartheta \wedge \eta] \cdot \mathbf{p}\right) \left(V[\eta] \cdot \mathbf{p}\right) = \left(V[\theta \wedge \eta] \cdot \mathbf{p}\right) \left(V[\vartheta \wedge \eta] \cdot \mathbf{p}\right). \quad (2)$$

Suppose now that s statements of the form $SIN(\theta_j, \vartheta_j|\eta_j)$ are present in the knowledge base, indicating that θ_j and ϑ_j are strongly independent given η_j. Each one of these independent statements implies a non-linear expression (2), given the positivity assumption and the fact that multilinear programs have global optima in the boundary of the feasible region [24]. Thus s *polynomial* equalities of the form (2) must be added to problem (1). We can avoid the fractional term in the objective function without changing the basic properties of the problem:

Theorem 1 *Problem (1) with additional constraints (2) is equivalent to*

$$\min_{\mathbf{p}'} / \max_{\mathbf{p}'} \left(V[\phi^* \wedge \varphi^*] \cdot \mathbf{p}' \right), \tag{3}$$

subject to constraints (2), $V[\varphi^*]\cdot\mathbf{p}' = 1$, $p_k \geq 0$ *for all* k, $(V[\phi_i \wedge \varphi_i] - l_i V[\varphi_i])\cdot\mathbf{p}' \geq 0$ *and* $(V[\phi_i \wedge \varphi_i] - u_i V[\varphi_i]) \cdot \mathbf{p}' \leq 0$, *for all* $i = 1, \ldots, m$.

Proof. Introduce a new variable t such that $t^{-1} = (V[\varphi^*] \cdot \mathbf{p})$. Consider a new vector $\mathbf{p}' = \mathbf{p}t$. The objective function becomes $(V[\phi^* \wedge \varphi^*] \cdot (\mathbf{p}'/t))\, t = (V[\phi^* \wedge \varphi^*] \cdot \mathbf{p}')$. The constraint $t^{-1} = (V[\varphi^*] \cdot \mathbf{p}'/t)$ becomes $V[\varphi^*] \cdot \mathbf{p}' = 1$. As for the strong independence constraints, note that constraints (2) remain the same (but over $\mathbf{p}'$ instead of $\mathbf{p}$). The same applies to the remaining linear constraints. □

Unlike geometric programming problems, constraints (2) lead to nonconvex primal and dual programs. Existing solution methods produce sequences of sub-problems that eventually contain only the global optimum, using either branch-and-bound or cutting-plane techniques [25–29]. The algorithms of Maranas and Floudas [27], and Gochet and Smeers [25] produce convex nonlinear sub-problems, while Sherali and Adams' algorithm produces linear sub-problems [28].

The characteristics of Sherali and Adams' branch-and-bound algorithm make it particularly suitable for probabilistic logic. As the sub-problems generated by this method are linear programs, column generation techniques can be applied to them — and column generation techniques are necessary to handle large scale problems [5]. The idea of Sherali and Adams' algorithm is to replace products of variables by new "artificial" variables, and to solve the resulting linear problem. The algorithm iterates by branching over the range of variables whenever necessary, until each artificial variable is close enough to its corresponding product.

The probabilistic satisfiability problem is NP-Complete [4]. The inclusion of independence statements $SIN(\theta, \vartheta|\eta)$ makes it harder:

Theorem 2 *Problem (1) with additional constraints (2) is NP*$^{\mathrm{PP}}$*-Hard.*

Proof. We can reduce a binary credal network belief updating problem [30] to this problem, naming each node of the network as a formula ϕ_i, specifying each local probability constraint of the network $P(\phi_i|\mathrm{pa}(\phi_i))$ as a probabilistic logic constraint and inserting (a polynomial number of) independence statements $SIN(\phi_i, \phi_j|\phi_k)$ between any two nodes ϕ_i and ϕ_j that are separated by ϕ_k in the network. As the binary credal network belief updating problem is NP$^{\mathrm{PP}}$-Complete, this theorem follows. □

4 Graph-theoretic representations: PPL networks

The flexibility of propositional probabilistic logic with independence is attractive but comes at a price in computational complexity. Besides, a language that is too general and unstructured may overwhelm users with too many possible options. In this section we explore situations where assessments and judgements in probabilistic logic can be compactly organized using graphs — following the practical success of several statistical models based on graphs, such as Bayesian and Markov networks [31].

In Bayesian networks and their many variants, the underlying graphs serve simultaneously as an encoding for independence relations and for assessments. In the context of probabilistic logic it seems reasonable to aim at representations that can accommodate both probabilistic and purely logical assessments. As logical constraints have not "direction," we consider graphs $\mathcal{G}$ where each node is associated with a proposition, and containing directed and undirected edges. The undirected edges are clearly intended to represent logical constraints, as in constraint and mixed networks [32].

Our requirement about the graph $\mathcal{G}$ is that it is a *chain graph*; that is, it does not have any directed cycles [33]. The semantics of the graph is given by the following *Markov condition* over all subsets $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{S}$ of nodes of $\mathcal{G}$ [33, p. 76]: If $\mathbf{S}$ separates $\mathbf{A}$ and $\mathbf{B}$ in the smallest ancestral set containing $\mathbf{A} \cup \mathbf{B} \cup \mathbf{S}$, then $\mathbf{A}$ and $\mathbf{B}$ are independent conditional on $\mathbf{S}$. This condition is equivalent to various other "local" properties when probability values are all nonzero, an assumption we cannot make in the presence of logical constraints. We will always assume that *conditional* probabilities are computed for conditioning events that have positive probability. The assumption of acyclicity and the Markov condition are consistent with the representation of logical constraints by undirected edges. For ease of exposition, we also assume that logical constraints contain only two propositions each.

We call the resulting structures *binary PPL networks*. A similar scheme of assessments has been considered previously by Campos and Cozman [34], where a directed acyclic graph is associated with propositions and arbitrary assessments over logical formulas. Their resulting model is not as convenient as the one developed in this paper; it is a bit restrictive in its reliance on directed acyclic graphs, and it is too liberal in the assessments it accepts, leading to computational difficulties.

There are several algorithms that compute probabilities in standard, probabilistic chain graphs [33]. Algorithms developed for Bayesian networks can be easily adapted to chain graphs, and there is an extensive literature on the former. A particularly simple algorithm is *variable elimination* [35, 36]. The purpose of variable elimination is to efficiently compute an expression such as $\sum_{\mathbf{X}} \prod_i P(X_i|\mathrm{pa}(X_i))$, where $\mathbf{X}$ is a set of random variables X_i. In fact variable elimination can be directly applied to inference in chain graphs, where each term in the inner product may be a probability distribution or simply an unnormalized potential. The variable elimination algorithm has been recently employed to conduct inferences in *credal networks*; that is, graph-theoretical structures that are similar to Bayesian networks but where a random variable X_i may be associated to a *set* of probability distributions $K(X_i|\mathrm{pa}(X_i))$ [37, 38]. The idea is to first run variable elimination "symbolically" and store the intermediate expressions in the sum/product. These expressions form a multilinear program that is then solved.

The same idea can be applied to our current setting, where one may wish to compute lower and upper conditional probabilities. This result is reached by an algorithm of two stages. In the first stage, the PPL network is transformed in a Boolean credal network that encodes the dependence structure of the original model. In the second stage a modified version of Campos and Cozman algorithm [37] is used to compute the desired interval.

A few definitions are useful in the remainder of the paper. A *Boolean credal network* is a triple $(\mathcal{G}, \mathbb{X}, \mathbb{K})$ where $\mathcal{G}$ is a directed acyclic graph with each node associated to a Boolean random variable of $\mathbb{X}$ and $\mathbb{K}$ is a collection of credal sets. Arcs represent direct dependencies between variables in $\mathbb{X}$ and nodes are associated with locally specified credal sets $K(X_i|\mathrm{pa}(X_i))$ [39]. We assume that this structure satisfies the following *Markov condition*: every variable is *strongly* independent of its nondescendants nonparents given its parents. Given a credal network, an event of interest $\{X_q = i\}$ and a set of evidences $\mathbb{E}$, the belief updating procedure aims at computing the limits of a interval for $P(X_q = i|\mathbb{E})$. These limits are called the lower and upper probabilities of $\{X_q = i\}$ given $\mathbb{E}$. Currently, there are several algorithms for computing these values exactly [40, 37], although the time complexity of the problem has motivated the utilization of approximate algorithms [41–43]. Now, a *binary constraint network* [44] is a triple $(\mathcal{H}, \mathbb{X}, \mathbb{C})$ where $\mathcal{H}$ is an undirected graph and $\mathbb{C}$ is denotes a set of binary constraints on pairs of Boolean variables in $\mathbb{X}$. Each binary constraint C in $\mathbb{C}$ is associated with an edge of $\mathcal{H}$. The usual *constraint satisfaction* problem is to determine a instantiation of the variables in $\mathbb{X}$ that is consistent with all constraints [45].

A binary PPL network is a mixture of both credal and constraint networks.

Definition 1 *Let $\mathbb{X}$ be a set of propositional variables and $\mathbb{C}$ a set of binary logical constraints on variables in $\mathbb{X}$. A binary PPL network is a quadruple $(\mathcal{M}, \mathbb{X}, \mathbb{K}, \mathbb{C})$, where $\mathcal{M}$ is a chain graph with arcs encoding conditional probabilities (through local credal sets) and edges encoding binary logical constraints between its variables; that is, each node X_i is associated to a collection of local credal sets $K(X_i|\mathrm{pa}(X_i)) \in \mathbb{K}$ and each undirected edge $E = (X_i, X_j)$ is associated with a binary logical constraint C of $\mathbb{C}$ with probability of being true defined by a credal set.*

The Figure 1 illustrates the structure of a PPL network. Nodes represent propositional variables and arcs denote direct conditional dependency.

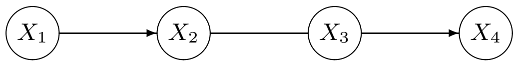

Fig. 1. Example of a binary PPL network.

We describe here the two stage procedure for belief updating in binary PPL networks. Suppose that the network of Figure 1 has the following probability distributions associated to it: $P(x_1) = 0.4$; $P(\overline{x_1}) = 0.6$; $P(x_2|x_1) = 0.3$; $P(\overline{x_2}|x_1) = 0.7$; $P(x_2|\overline{x_1}) = 0.5$; $P(\overline{x_2}|\overline{x_1}) = 0.5$; $P(x_4|x_3) = 0.2$; $P(\overline{x_4}|x_3) = 0.8$; $P(x_4|\overline{x_3}) = 0.6$; $P(\overline{x_4}|\overline{x_3}) = 0.4$. Suppose also that it was not possible to produce a Bayesian network because the knowledge engineer could not elicit $P(X_3|X_2)$ or $P(X_2|X_3)$; however suppose it was possible to estimate an interval to $P(X_3)$ represented by the constraint $0.3 \leq P(X_3 = x_3) \leq 0.6$. We will write $P(x_3)$ instead of $P(X_3 = x_3)$ where possible. Additionally, suppose that external data allowed to state that $P(\phi)$ is equal to or greater than 0.252, where $\phi = (x_2 \wedge x_3)$. This constraint is associated to the edge that connects X_2 and X_3 and is not part of a usual credal network.

Now assume we need to compute the maximum possible value of $P(x_4|x_1)$. In the first stage, we transform the binary PPL network into an auxiliary credal network that has the same nodes and directed arcs as the original network, but each undirected edge (X_i, X_j) is replaced by a new artificial node Y, child of X_i and X_j. The local credal sets $K(Y|X_i, X_j)$ are defined through the original credal sets associated to the logical constraint. Figure 2 shows this auxiliary network. C_1 is the artificial node.

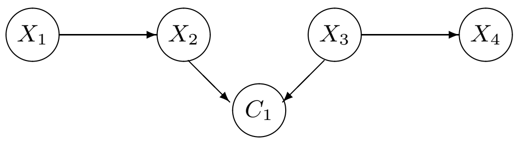

Fig. 2. The auxiliary credal network.

In the second stage, a modified version of the multilinear programming algorithm proposed by Campos and Cozman [37] for inference in credal networks is executed.

The multilinear programming problem becomes $\min / \max P(x_4|x_1)$, subject to $P(x_4, x_1) = P(x_4|x_1)(P(x_4, x_1) + P(\overline{x_4}, x_1))$ and

$$\begin{aligned}
P(X_4, x_1) &= P(c_1, X_4, x_1) + P(\overline{c_1}, X_4, x_1), \text{ where } X_4 \in \{x_4, \overline{x_4}\}, \\
P(C_1, X_4, x_1) &= P(C_1, x_1|x_3)P(X_4|x_3)P(x_3) + P(C_1, x_1|\overline{x_3})P(X_4|\overline{x_3})P(\overline{x_3}), \\
&\quad \text{where } C_1 \in \{c_1, \overline{c_1}\}, X_4 \in \{x_4, \overline{x_4}\}, \\
P(C_1, X_1|X_3) &= P(C_1|x_2, X_3)P(X_1, x_2) + P(C_1|\overline{x_2}, X_3)P(X_1, \overline{x_2}), \\
&\quad \text{where } C_1 \in \{c_1, \overline{c_1}\}, X_1 \in \{x_1, \overline{x_1}\}, X_3 \in \{x_3, \overline{x_3}\}, \\
P(X_1, X_2) &= P(X_2|X_1)P(X_1), \text{ where } X_1 \in \{x_1, \overline{x_1}\}, X_2 \in \{x_2, \overline{x_2}\}
\end{aligned}$$

and the linear constraints defining the local credal sets. Besides the pure probabilistic assessments, we have $P(c_1) \geq 0.252$ which implies the following additional constraints:

$$\begin{aligned}
P(c_1) &= P(c_1, x_3) + P(c_1, \overline{x_3}) \geq 0.252, \\
P(c_1, X_3) &= P(c_1, x_1|X_3)P(X_3) + P(c_1, \overline{x_1}|X_3)P(X_3) \text{ where } X_3 \in \{x_3, \overline{x_3}\}.
\end{aligned}$$

Our implementation promptly produces $P(x_4|x_1) = 0.36$. If the constraint between X_2 and X_3 (the one that implied $P(c_1) \geq 0.252$) is discarded, then we get the interval $\min P(x_4|x_1) = 0.36$ and $\max P(x_4|x_1) = 0.48$. The following result is relevant:

Theorem 3 *The inference in a binary PPL network is NP^{PP}-Complete.*

Sketch of Proof. Hardness comes directly from the fact that a binary PPL network is a extension to the Boolean credal network, so we can trivially reduce the belief updating problem in credal networks to inference in PPL networks. Pertinence is achieved because, when we fix the vertices of the credal sets, we obtain a standard Bayesian network. Then the PP oracle is enough to verify the pertinence. □

5 Conclusion

In this paper we have investigated algorithms for probabilistic logic in the presence of statements of strong independence. We would like to stress the following contributions of the paper. We presented algorithms based on Sherali and Adams' algorithm that produce inferences for large models (compared to the models that can be handled by existing algorithms). We focused on exact inferences, hoping that approximation methods will follow in time. Second, we explored graphical models (in the context of strong independence) that can be used to build large multivariate models in a compact manner. Future work can follow several paths: first order logic constructs, conditioning on zero probability events, coherency-based inference, and general improvement on the efficiency of inference algorithms.

In closing, it is appropriate to discuss the results of this paper in a broader perspective. The strategy here is to combine "classic" probabilistic logic with independence, retaining the freedom usually associated with the former. We do not require that enforced independence relations be specified in any particular fashion; this is to be contrasted with proposals usually labeled "probabilistic relational." In these proposals the idea is to combine logic and probability by restricting the language, so as to obtain Bayesian networks for inference [46–51]. While independence-free probabilistic logic may be *too loose*, probabilistic relational models may be *too strict*, as they demand a certain number of independence relations in a specific order. Probabilistic logic with independence seems to be a sensible middle ground; one can either move in the direction of complete generality, or one can build rather specific models. As an example, PPL networks are models that stay between fully general probabilistic logic and probabilistic relational models — accepting that some structure is necessary, but rejecting that a single recipe can be used in every knowledge base.

Acknowledgements

This work has received substantial support from FAPESP through grant 04/09568-0. We thank Barbara Vantaggi and Thomas Lukasiewicz for helping with our understanding of independence in probabilistic logic. The work has also received partial support from CNPq (through grant 3000183/98-4) and HP Brasil R&D.

References

1. Halpern, J.Y.: An analysis of first-order logics of probability. In: International Joint Conference on Artificial Intelligence, Detroit, USA (1989) 1375–1381
2. Halpern, J.Y.: Reasoning about Uncertainty. MIT Press (2003)
3. Nerode, A., Shore, R.A.: Logic for applications. Second edn. Springer Verlag, Ney York (1997)
4. Georgakopoulos, G., Kavvadias, D., Papadimitriou, C.H.: Probabilistic satisfiability. Journal of Complexity **4** (1988) 1–11

5. Hansen, P., Jaumard, B.: Probabilistic satisfiability. Technical report, École Polytechique de Montréal (1996)
6. Boole, G.: The Laws of Thought. Dover edition (1958)
7. Hailperin, T.: Best possible inequalities for the probability of a logical function of events. American Mathematical Monthly **72** (1965) 343–359
8. Nilsson, N.J.: Probabilistic logic. Artificial Intelligence **28** (1986) 71–87
9. de Finetti, B.: Theory of Probability, vol. 1-2. Wiley, New York (1974)
10. Bruno, G., Gilio, A.: Applicazione del metodo del simplesso al teorema fondamentale per le probabilita nella concezione soggettivistica. Statistica **40** (1980) 337–344
11. Coletti, G., Scozzafava, R.: Stochastic independence in a coherent setting. Annals of Mathematics and Artificial Intelligence **35** (2002) 151–176
12. Biazzo, V., Gilio, A., Lukasiewicz, T., Sanfilippo, G.: Probabilistic logic under coherence: Complexity and algorithms. In: International Symposium on Imprecise Probabilities and Their Applications, Shaker (2001) 51–61
13. Andersen, K.A., Hooker, J.N.: Bayesian logic. Decision Support Systems **11** (1994) 191–210
14. Coletti, G., Scozzafava, R.: Zero probabilities in stochastic independence. Information, Uncertainty, Fusion (2000) 185–196
15. Vantaggi, B.: Conditional Independence and Graphical Models in a Coherent Setting. PhD thesis, Università di Perugia, Italy (1999)
16. Lakshmanan, L.V.S., Sadri, F.: Probabilistic deductive databases. In: Symposium on Logic Programming. (1994) 254–268
17. Ng, R., Subrahmanian, V.S.: Probabilistic logic programming. Information and Computation **101(2)** (1992) 150–201
18. Lukasiewicz, T.: Probabilistic logic programming. In: European Conference on Artificial Intelligence. (1998) 388–392
19. Bacchus, F.: Representing and Reasoning with Probabilistic Knowledge. MIT Press, Cambridge, Mass. (1990).
20. Cozman, F.G.: Calculation of posterior bounds given convex sets of prior probability measures and likelihood functions. Journal of Computational and Graphical Statistics **8(4)** (1999) 824–838
21. Couso, I., Moral, S., Walley, P.: A survey of concepts of independence for imprecise probabilities. Risk, Decision and Policy **5** (2000) 165–181
22. de Campos, L., Moral, S.: Independence concepts for convex sets of probabilities. In: Conference on Uncertainty in Artificial Intelligence, San Francisco, Morgan Kaufmann (1995) 108–115
23. Walley, P.: Statistical Reasoning with Imprecise Probabilities. Chapman and Hall, London (1991)
24. Drenick, R.F.: Multilinear programming: Duality theories. Journal of Optimization Theory and Applications **72(3)** (1992) 459–486
25. Gochet, W., Smeers, Y.: A branch-and-bound method for reversed geometric programming. Operations Research **27(5)** (1979) 983–996
26. Horst, R., Tuy, H.: Global Optimization: Deterministic Approaches. Springer Verlag, Heidelberg (1996)
27. Maranas, C.D., Floudas, C.A.: Global optimization in generalized geometric programming. Computers and Chemical Engineering **21(4)** (1997) 351–370
28. Sherali, H.D., Adams, W.P.: A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems. Kluwer Academic Publishers, Dordrecht/Boston/London (1999)
29. Tuy, H.: Convex Analysis and Global Optimization. Volume 22 of Nonconvex Optimization and Its Applications. Kluwer Academic Publishers, Dordrecht (1998)

30. de Campos, C.P., Cozman, F.G.: The inferential complexity of Bayesian and credal networks. In: International Joint Conference on Artificial Intelligence. (2005) 1313–1318
31. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, San Mateo, California (1988)
32. Dechter, R., Mateescu, R.: Mixtures of deterministic-probabilistic networks and their AND/OR search space. In: Conference on Uncertainty in Artificial Intelligence, Arlington, Virginia, AUAI Press (2004) 120–129
33. Cowell, R., Dawid, P., Lauritzen, S., Spiegelhalter, D.: Probabilistic Networks and Experts Systems. Springer-Verlag, New York (1999)
34. de Campos, C.P., Cozman, F.G.: Belief updating and learning in semi-qualitative probabilistic networks. In: Conference on Uncertainty in Artificial Intelligence. (2005) 153–160
35. Zhang, N.L., Poole, D.: A simple approach to bayesian network computations. In: Canadian Conference on Artificial Intelligence. (1994) 171–178
36. Dechter, R.: Bucket elimination: A unifying framework for probabilistic inference. In Horvitz, E., Jensen, F., eds.: Conference on Uncertainty in Artificial Intelligence, San Francisco, Morgan Kaufmann Publishers (1996) 211–219
37. de Campos, C.P., Cozman, F.G.: Inference in credal networks using multilinear programming. In: Second Starting AI Researcher Symposium, Valencia, IOS Press (2004) 50–61
38. Cozman, F.G., de Campos, C.P., Ide, J.S., da Rocha, J.C.F.: Propositional and relational Bayesian networks associated with imprecise and qualitative probabilistic assessments. In: Conference on Uncertainty in Artificial Intelligence, Banff, AUAI Press (2004) 104–111
39. Cozman, F.G.: Credal networks. Artificial Intelligence **120** (2000) 199–233
40. da Rocha, J.C.F., Cozman, F.G.: Inference in credal networks with branch-and-bound algorithms. In: International Symposium on Imprecise Probability and Their Applications, Lugano, Switzerland, Carleton Scientific (2003)
41. da Rocha, J.C.F., Cozman, F.G., de Campos, C.P.: Inference in polytrees with sets of probabilities. In: Conference on Uncertainty in Artificial Intelligence, New York, Morgan Kaufmann (2003) 217–224
42. Ide, J.S., Cozman, F.G.: Approximate inference in credal networks by variational mean field methods. In: International Symposium on Imprecise Probability and Their Applications. (2005) 203–212
43. Cozman, F.G., de Campos, C.P.: Local computation in credal networks. In: Workshop on Local Computation for Logics and Uncertainty, Valencia, IOS Press (2004) 5–11
44. Ladkin, P.B., Maddux, R.: On binary constraint problems. Journal of the ACM **41(3)** (1994) 435–469
45. van Beek, P., Dechter, R.: On the minimality and global consistency of row-convex constraint networks. Journal of the ACM **42(3)** (1995) 543–561
46. Gettor, L., Friedman, N., Koller, D., Taskar, B.: Learning probabilistic models of relational structure. In: International Conference on Machine Learning. (2001) 170–177
47. Glesner, S., Koller, D.: Constructing flexible dynamic belief networks from first-order probabilistic knowledge bases. Symbolic and Quantitative Approaches to Reasoning with Uncertainty (1995) 217–226
48. Haddawy, P., Suwandi, M.: Decision-theoretic refinement planning using inheritance abstraction. In: AAAI Spring Symposium Series: Decision-Theoretic Planning. (1994)
49. Jaeger, M.: Relational Bayesian networks. In: Conference on Uncertainty in Artificial Intelligence, San Francisco, Morgan Kaufmann (1997) 266–273
50. Ngo, L., Haddawy, P.: Answering queries from context-sensitive probabilistic knowledge bases. Theoretical Computer Science **171(1–2)** (1997) 147–177
51. Poole, D.: Probabilistic Horn abduction and Bayesian networks. Artificial Intelligence **64** (1993) 81–129