International Journal of Approximate Reasoning 58 (2015) 57–70

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

www.elsevier.com/locate/ijar

Probabilistic satisfiability and coherence checking through integer programming

Fabio G. Cozman *, Lucas Fargoni di Ianni

Universidade de São Paulo, Av. Prof. Mello Moraes, 2231, São Paulo, SP, Brazil

ARTICLE INFO

Article history:
Received 13 May 2014
Received in revised form 1 September 2014
Accepted 11 September 2014
Available online 18 September 2014

Keywords:
Probabilistic logic
Probabilistic satisfiability
Coherence
Integer programming
Phase transitions

ABSTRACT

This paper presents algorithms, both for probabilistic satisfiability and for coherence checking, that rely on reduction to integer programming. That is, we verify whether probabilistic assessments can be satisfied by standard probability measures (Kolmogorovian setting) or by full conditional probabilities (de Finettian coherence setting), and in both cases verify satisfiability or coherence using integer programming techniques. We present an empirical evaluation of our method, the results of which show evidence of phase transitions.

1. Introduction

The analysis of arguments that combine propositions and probabilities has deserved attention for quite some time. For instance, in Boole's work we find interesting examples such as:

> The probability that it thunders upon a given day is p, the probability that it both thunders and hails is q, but of the connexion of the two phenomena of thunder and hail, nothing further is supposed to be known. Required the probability that it hails on the proposed day. [13, Chapter XVIII, Ex. 1]

Here we have propositions A and B, assessments $\mathbb{P}(A) = p$ and $\mathbb{P}(A \wedge B) = q$. Boole asks for $\mathbb{P}(B)$ and obtains the tight interval $[q, 1 - (p - q)]$. There is a probability measure that *satisfies* the assessments; for this reason, they are *coherent*.

Suppose we have atomic propositions $\{A_j\}_{j=1}^n$ and propositional sentences $\{\phi_i\}_{i=1}^M$ involving those atomic propositions. We may associate one or more of these sentences with probabilities, writing for instance $\mathbb{P}(\phi_i) = \alpha_i$. As detailed later, to establish semantics for these assessments we consider a probability measure over the set of truth assignments. The *Probabilistic Satisfiability (PSAT)* problem is to determine whether it is possible to find a probability measure over truth assignments such that all assessments are satisfied [25,28,31,33,34,37]. When assessments involve conditional probabilities such as $\mathbb{P}(A|B) = \alpha$, there are two paths to follow. Kolmogorovian probability theory reduces such assessments to ratios of probabilities. The other path is to use de Finetti's theory of coherent probabilities, where full conditional probabilities are used to interpret conditional assessments [19,23,58]. The *Coherence Checking (CCHECK)* problem is to determine whether it is possible to find a full conditional probability that satisfies all assessments, without requiring that assessments are over an algebra or any other structure [5,6]. Coherence checking has been explored in a variety of settings; a typical example is:

* Corresponding author.

http://dx.doi.org/10.1016/j.ijar.2014.09.002
0888-613X/

A doctor considers three possible diagnoses, H_1 (ileum), H_2 (peritonitis), and H_3 (appendicitis with local peritonitis), with the logical condition that $H_3 \to (\neg H_1) \wedge H_2$, and assessments $\mathbb{P}(H_1) = 1/2$, $\mathbb{P}(H_2) = 1/5$, and $\mathbb{P}(H_3) = 1/8$. Note that diagnoses do not constitute a partition of the certain event. The assessments are *coherent* in that they are satisfied by at least a probability measure. The doctor now considers E (no pain in abdomen), notes that $H_3 \to \neg E$, and declares $\mathbb{P}(E|H_2) = 2/5$ and $\mathbb{P}(E|\neg H_2) = 1/8$. Now the whole set of assessments fails to be coherent.[1]

Probabilistic satisfiability and coherence checking are central problems in reasoning under uncertainty. They serve not only as a foundation for logical and probabilistic inference, but as a basis for probabilistic rules [50], and as an initial necessary step in the understanding of combinations of first-order logic and probabilities [35,46,51].

The most direct way to solve a PSAT problem is to write it down as a linear feasibility problem [33]. The difficulty is that the resulting linear program may be too large; for n propositions, we must build a matrix with up to 2^n columns. When conditional probabilities are present, coherence checking may require sequences of such linear programs. To avoid dealing with exponentially many columns, one may resort to column generation techniques [40], to inference rules that capture probabilistic relationships [6], or even to combinations of column generation and inference rules [38]. There is also a different approach to probabilistic satisfiability that reduces it to logical satisfiability [4,26]. Overall, results in the literature save computations by applying increasingly complicated methods.

In this paper we present another approach to probabilistic satisfiability and coherence checking, where these problems are turned into integer programs. Our basic algorithm for PSAT is rather concise and easy to implement when a linear solver is available. In fact, our goal is to present methods that can be applied to medium sized problems, with say some 20 to 200 atomic propositions, by exploiting the fact that integer programming technology has improved dramatically in recent years. So, instead of explicitly resorting to inference rules and column generation, our methods simply outsource such schemes to the linear solver, as top solvers do apply sophisticated heuristics and numerical stabilization internally. We show that our techniques can be easily extended to expectation assessments, and describe ways to reduce coherence checking to (sequences of) integer linear programs.

As our experiments show, integer programming techniques are not yet capable of beating the fastest methods in the literature for large problems, but they offer a robust basis for PSAT and CCHECK. Using our implementation we study the issue of phase transition in probabilistic satisfiability, showing evidence of interesting phenomena in PSAT.

Section 2 summarizes necessary background in satisfiability and probability satisfiability. Our basic algorithm for probabilistic satisfiability is described in Section 3. We consider extensions of probabilistic satisfiability in Section 4, and then study coherence checking in Section 5. Implementation and experiments, with a discussion of phase transitions, are presented in Section 6.

2. SAT and PSAT

Consider n atomic propositions A_j and M sentences ϕ_i in propositional logic involving those atomic propositions. A truth assignment is an assignment of truth values (True or False) to each atomic proposition, that induces an assignment of truth values for all sentences involving the atomic propositions. If a truth assignment ω is such that sentence ϕ is True, write $\omega \models \phi$. The Satisfiability (SAT) problem is to determine whether or not there exists a truth assignment to all atomic propositions such that all sentences evaluate to True [18,30].

If every sentence ϕ_i is a conjunction of clauses, then we have a SAT problem in *Conjunctive Normal Form (CNF)*. A SAT problem in CNF is a k-SAT problem when each clause has k literals (note that literals may appear more than once in a clause, so in fact we can have up to k distinct literals). The 2-SAT problem has a polynomial solution, while k-SAT is NP-complete for $k > 2$.

For a fixed n, m and k, one may generate a random k-SAT with n propositions and m clauses, as follows. For each one of the m clauses: select k propositions at random, and for each proposition produce a literal that may be negated or not, with probability half. There has been intense study of *phase transition* phenomena in random k-SAT; that is, the observed fact that for small values of m/n the probability that a random k-SAT is satisfiable tends to one as n grows (at fixed m/n), while for large values of m/n the probability that a random k-SAT is satisfiable tends to zero as n grows [30]. Moreover, in the regions where satisfiability has probability approaching zero or one we observe that generated random k-SAT problems can be easily solved, while in the transition between the two regions we find hard problems.

Suppose that some sentences, say ϕ_1 to ϕ_q, for $q \leq M$, are associated with probabilities through *assessments* of the form $\mathbb{P}(\phi_i) \bowtie \alpha_i$, where $\bowtie$ is one of $\geq, =, \leq$. The semantics of such an assessment is as follows. Take the set of 2^n truth assignments that can be generated for the n propositions. A probability measure $\mathbb{P}$ over this set satisfies the assessments if, for each assessment $\mathbb{P}(\phi_i) \bowtie \alpha_i$,

$$\sum_{\omega \models \phi_i} \mathbb{P}(\omega) \bowtie \alpha_i. \tag{1}$$

[1] The example is due to Coletti and Scozzafava [21, Example 1], and appears edited here. Failure of coherence can be verified as follows. We must have $\mathbb{P}(H_2) = 1/5 = \mathbb{P}(H_2 \wedge E) + \mathbb{P}(H_2 \wedge \neg E) \geq \mathbb{P}(H_2 \wedge E) + \mathbb{P}(H_2 \wedge \neg E \wedge (\neg H_1 \wedge H_3))$; but $\mathbb{P}(H_2 \wedge E) = 2/25$ (multiplying $\mathbb{P}(H_2) = 1/5$ and $\mathbb{P}(E|H_2) = 2/5$) and $\mathbb{P}(H_2 \wedge \neg E \wedge (\neg H_1 \wedge H_3)) = \mathbb{P}(H_3) = 1/8$ (because $H_2 \wedge \neg E \wedge (\neg H_1 \wedge H_3)$ is equivalent to H_3 as $H_3 \to (\neg H_1) \wedge H_2$ and $H_3 \to \neg E$).