Elimination of Boolean variables for probabilistic coherence

M. Baioletti, A. Capotorti, S. Tulipani, B. Vantaggi

Abstract  In this paper we deal with the computational complexity problem of checking the coherence of a partial probability assessment (called CPA). The CPA problem, like its analogous PSAT, is NP-complete so we look for an heuristic procedure to make tractable reasonable instances of the problem. Starting from the characteristic feature of de Finetti’s approach (i.e. the explicit distinction between the probabilistic assessment and the logical relations among the sentences) we introduce several rules for a sequential “elimination” of Boolean variables from the domain of the assessment. The procedure resembles the well-known Davis-Putnam rules for the satisfiability, however we have, as a drawback, the introduction of constraints (among real variables) whose satisfiability must be checked. In simple examples we test the efficiency of the procedure respect to the “traditional” approach of solving a linear system with a huge coefficient matrix built from the atoms generated by the domain of the assessment.

Key words  coherent probability assessments, PSAT problem, NP-complete problems, elimination of Boolean variables

1 Introduction

G. Boole, in his book “An investigation of the laws of thought” [1], suggests and deals with the problem of a probabilistic assessment. The general problem is expressed as follows: Given the probabilities of any events of whatever kind, to find the probability of some other event connected with them. This problem is strictly connected with the satisfiability problem, called by Boole conditions of possible experience. Given some propositional logic sentences $S_1, \ldots, S_m$ on the variables $X_1, \ldots, X_n$ and given the real numbers $p_1, \ldots, p_m$ belonging to the real interval $[0, 1]$, we want to check if there exists a probability distribution $P$ over the statements’ algebra (modulo logical equivalence) such that $P(S_i) = p_i$ for $i = 1, \ldots, m$.

A restriction of this problem to the class of statements called clauses, that are simply disjunctions of literals, has been recently [12] called PSAT and it was proved to be in NP. Since it contains the SAT problem of satisfiability of sets of Boolean clauses, it is a NP-complete problem.

Moreover also the subproblem 2PSAT, the restriction of PSAT to clauses with at most two literals, is a NP-hard problem. This goes against the result of 2SAT, the satisfiability of Boolean clauses with at most two literals, that is in P-time (see [11]).

The probabilistic satisfiability problem was also introduced later by de Finetti [8] as a basic notion for the subjective probabilistic approach. The two point of view of the problem are strongly related, but one of the main difference of de Finetti’s approach is the explicit distinction between the probabilistic assessment and the logical relations among the sentences.

More precisely, de Finetti’s probabilistic satisfiability problem, called check of coherence, is posed directly for an assessment $p_1, \ldots, p_n$ on a set of literals $E_1, \ldots, E_n$ (called events) equipped with logical constraints (e.g. $E_i$ implies $E_j$, $E_k$ is incompatible with $E_i$, etc.). Hence, this framework is more general than the classical probabilistic theory where the probability assessment is given on a structured set (algebra or $\sigma$-algebra). For this reason the usual Kolmogorov axioms are not sufficient to guarantee the soundness of such assessment. De Finetti dealt also with Boole’s general problem by introducing the Fundamental Theorem of Probability [8] that allows to compute the lower and upper bounds of a coherent probability for any event outside the initial set. This technique is useful to make inference on new (not taken in consideration before) aspects of a practical problem, so an user (decision maker, field expert, etc.) has a valid guideline to perform a dynamic evaluation.

Boole’s probability approach and his solutions to the general problem were re-discovered and refreshed by Hailperin [15, 16]. After his work, others [10, 9] generalized the problem introducing a logic to deal with probability; the tools of this logic (see [9]) are polynomials with integer coefficients, and inequalities among them represent constraints for the probabilities of the starting events. Also in this framework, not worse than for PSAT, the satisfiability problem for sets of sentences, expressed in such logics, is in NP and hence it is NP-complete.

Starting from Boole, both de Finetti’s check of coherence and Boole’s general problem are solved searching non-negative solutions of a linear system

$$Ay = p$$

(1)
where \( y \) is a column vector of unknowns with \( 2^n \) elements (each one associated to one of the atoms in the free algebra \( \mathcal{F}(X_1, \ldots, X_n) \) generated by the Boolean variables \( X_1, \ldots, X_n \); \( p \) is the probability column vector \((1, p_1, \ldots, p_m)^T\) on \( S_0, \ldots, S_m \) where \( S_0 \) the sure event; 
\( A = [a_{ij}] \) is a \((m + 1) \times 2^n\) matrix with \( a_{ij} = 1 \) iff the \( j \)-th atom (within a specified numeration) of \( \mathcal{F}(X_1, \ldots, X_n) \) is included in the sentence \( S_i \) (i.e. the truth assignment \( v : \{X_1, \ldots, X_n\} \rightarrow \{0, 1\} \) which makes true the \( j \)-th atoms makes true also the sentence \( S_i \)); \( a_{ij} = 0 \) otherwise.

Note that the first row of \( A \) has all the components equal to \( 1 \), so the sum of the unknowns in \( y \) must be equal to \( 1 \).

Since the system (1) has an exponential columns number respect to the dimension of the input, starting from Hailperin, several authors [19, 18, 21] have proposed algorithms based on linear programming techniques to check the consistency of (1) or to find out its solutions. These techniques start from the well known result that, if (1) is consistent then there is some solution with at most \( n + 1 \) non-zero components.

Also de Finetti’s approach has been developed by many authors (see for example [4, 6, 7, 13, 20, 22, 23, 24]), but they focused mainly on the probabilitistic aspects and on its applications instead of the complexity problem. A first attempt in this direction is in [2] where assessment on particular sets of events are investigated and characterized without the generation of atoms [i.e. without introducing linear systems like (1)].

In this paper, following de Finetti’s guide-line, we will deal with the coherence of a probability assessment on an arbitrary set of events \( E_1, \ldots, E_n \). We will denote this problem with CPA (Coherence of a Probability Assessment). As written before, in this framework the Boolean relationships among the events \( E_1, \ldots, E_n \) are given, so we have a generator system for the ideal \( J \) of the free Boolean algebra \( \mathcal{F}(X_1, \ldots, X_n) \) which is the kernel of the unique morphism \( \hat{d} : \mathcal{F}(X_1, \ldots, X_n) \rightarrow \mathcal{B} \) such that \( \hat{d}(X_i) = E_i \).

The main difference between this paper and the others cited before is that we will not build at the beginning the huge linear system (1), but we will erase the Boolean variables from the domain with a procedure which resembles the well known Davis-Putnam rules for the satisfiability of a set of Boolean clauses.

However, our splitting rule is in a sense reversed, since it reduces the satisfiability of a given system to the satisfiability of both the derived systems. Moreover, we have, as a drawback, the introduction of constraints (among real variables) whose satisfiability must be checked.

We will propose an algorithm that, like Davis-Putnam procedure, does not always guarantee a solution in reasonable time. Nevertheless we expect good performance, in particular for 2CPA, i.e. the CPA problem with binary clauses. This paper will be developed with simple examples to explain better our methodology. An actual implementation will follow in a future work where we will introduce and discuss some simplification rules with the aim of reducing the search space. Our intent will be also to test the performance and to compare the algorithm with other methods known in literature [17, 19].

2 Preliminary notions and problem description

In this paper we use the so called dual clauses: they are nothing more than simple conjunctions of Boolean literals. To convert the usual clauses in their duals it needs only to change each proposition with its negation (or complement) and each probability value \( p \) with \( 1 - p \). We will denote a dual clause, containing the literals \( L_1, \ldots, L_r \), simply by \( L_1L_2 \ldots L_r \). So, when \( X_i \) is one of our Boolean variable, we use (when it will not cause misunderstanding) \( X_i \) and \( X'_i \) both as literals and as unitary clauses.

We denote with \( V \) a finite set of Boolean variables, usually \( V = \{X_1, \ldots, X_n\} \). When \( C \) denotes a set of dual clauses in the variables \( V \), the satisfiability problem for \( C \) is to find a truth assignment \( v \in 2^V \) (2 denotes the algebra \( \{0, 1\} \)) such that the truth value \( v(C) \) is equal to \( 0 \) for all \( C \in C \). In other words, each clause \( C \) must be understood as an equation to be solved in \( \{0, 1\} \) and so it must be seen as a function of \( n \) variables \( C : 2^V \rightarrow \{0, 1\} \). Hence the satisfiability problem is reduced to find a common solution of all the equations arising from all the clauses in \( C \).

Generally speaking, a probabilistic satisfiability problem is given by a set of \( n \) elements \( E_1, \ldots, E_n \) (called events) endowed with a probabilistic assessment \( p_1, \ldots, p_n \in [0, 1] \). The aim is to find out if there exists a probability distribution \( Q \) over the Boolean algebra \( \mathcal{B} \) generated by the \( n \) events, such that \( Q(E_i) = p_i \) for \( i = 1, \ldots, n \). Note that this formulation is ill-posed since the possible relations among the generators \( E_1, \ldots, E_n \), are not considered. This lack can be solved by introducing the morphism

\[
\hat{d} : \mathcal{F}(V) \rightarrow \mathcal{B}
\]

of the free algebra on the variables \( V = \{X_1, \ldots, X_n\} \), induced by the application

\[
\hat{d} : V \rightarrow \mathcal{B}
\]

\[
X_i \mapsto E_i.
\]

The ideal \( J \), kernel of the morphism \( \hat{d} \), contains all the relations among the generators. More precisely, \( \hat{d}(H) = 0 \) holds for all \( H \in J \). Note that a set of dual clauses \( C_1, \ldots, C_m \) can be chosen for the generators of \( J \) and \( E_1, \ldots, E_n \) satisfy these clauses in the algebra \( \mathcal{B} \) (i.e. \( \hat{d}(C_j) = 0, j = 1, \ldots, m \)).

The use of dual clauses is not restrictive because each logical relation among events can be expressed by some negation of tautology, i.e. a formula that must always evaluate to false. Taking for each formula its DNF normal form, all its disjuncts are dual clauses and the original formula evaluates to false iff all such clauses evaluate to false.

Now we may formalize correctly the CPA problem [8]:

CPA problem:
Given a finite set of events \( E_1, \ldots, E_n \), with logical relations among them expressed by the dual clauses \( C_1, \ldots, C_m \) (on the variables \( X_1, \ldots, X_n \)) and given the assessment \( E_i \mapsto p_i \), \( i = 1 \ldots n \), does there exist a probability distribution \( Q : \mathcal{B} \rightarrow [0, 1] \), on the Boolean algebra \( \mathcal{B} \) generated by \( E_1, \ldots, E_n \), such that \( Q(E_i) = p_i \) for \( i = 1, \ldots, n \)?