Probabilistic logic: the PSAT and CPA models

Extended Abstract

Daniele Pretolani*

February 2002

1 Introduction

Several attempts have been made to combine logic and probability theory within a common framework. Ultimately, the goal is to provide a formal setting for dealing with the *general problem* (or *inference problem*): given the probabilities of a certain set of events, derive probabilities for some other events that are logically connected with them. This problem is closely related to the *satisfiability problem*, where one has to check the consistency of a given assignment of probabilities.

The issue is not new, in fact, it dates back (at least) to George Boole [4] in the nineteenth century. Boole's satisfiability problem can be defined in the following terms: Given a set of logical sentences on a set of propositional variables, and given a set of probabilities assigned to the sentences, check if there exists a probability distribution on the space of *possible worlds* (i.e. truth assignments to variables) that is consistent with the assigned probabilities. More recently, Boole's approach has been rediscovered in the framework of *probabilistic logic* [9, 11], where the above problem is known as *probabilistic satisfiability* (PSAT); usually, PSAT is defined on CNF formulas, where logical sentences are *clauses*. Within probabilistic logic, the general problem is usually referred to as *entailment*, and can be defined as follows: based on a consistent probability assignment to a set of clauses, find consistent probability bounds for a new clause on the same set of variables.

A different satisfiability problem (called *checking of coherence*) has been introduced by de Finetti [7], and is strictly related to the basic principles of his work in probability theory. In this problem, we are given an assessment of probabilities to a set of *events*, related to each other by a set of *logical constraints* represented by

*Dipartimento di Matematica e Informatica, Università di Camerino, via Madonna delle Carceri, I-62032 Camerino (Italia). *e-mail* daniele.pretolani@unicam.it

logical sentences (clauses). This problem is usually referred to as *Coherence of Probability Assessment* (CPA). In his work, de Finetti also deals with the general problem, showing how to infer coherent probability bounds for other events outside the initial set.

Both PSAT and CPA can be considered as a probabilistic counterpart of the classical SAT problem. It can be remarked that the two problems show opposite features. In CPA, probabilities are associated with propositional variables (events), while clauses define the logical relations between events, and have no associated "uncertainty". In the PSAT problem, logical relations between variables are hidden, and uncertainty is associated with the clauses. Note that the corresponding general problems also consider different situations, i.e. new events (CPA) and new logical relations (PSAT).

As a matter of fact, it turns out that CPA is a particular case of the more general PSAT problem. Nevertheless, CPA shows peculiar aspects that are worth investigating. Here, we shall point out three of these aspects.

In Section 2, we shall consider the classical linear programming formulation of PSAT. This formulation can be dealt with by *column generation* techniques (see e.g. [8]), where the *pricing problem* is *Weighted Max SAT*. This general formulation can be simplified for the CPA problem, thus leading to a *Weighted Min SAT* pricing problem. The relations between these problems, and some of their implications, will be discussed throughout the paper.

In Section 3 we shall consider the *extended set covering* formulation of PSAT proposed in [1]. For particular classes of instances, PSAT reduces to solving a relaxation of this formulation, that is, a set of linear inequalities whose size is linear in the size of the instance. For the CPA problem, the formulation can be simplified: as a result, a wider class of instances can be reduced to an even simpler problem, i.e. checking that the given probabilities satisfy a (small) system of inequalities.

In Section 4 we shall point out the relations between CPA and *Horn Formulae with uncertainty* [2]. These formulae provide a computationally simple model for representing uncertainty in a knowledge base system, where *certainty factors* are associated with facts. Here, we shall show that CPA provides a semantic interpretation for this model, where certainty factors are interpreted as probability assessments.

Finally, some rather theoretical open questions will be discussed in Section 5.

It must be remarked that the relevance of the CPA problem goes far beyond the aspects outlined here. For example, CPA plays a key rôle in the probabilistic treatment of partial knowledge, in particular for what concerns *probabilistic inference* for *conditional events*, that can be reduced to a sequence of problems for unconditional events [5]. This in turn provides motivation for research on the algorithmic side. For example, effective algorithms, based on efficient reduction techniques, have been devised in [3].

2 Formulation and Column Generation

A *formula* in *Conjunctive Normal Form* (CNF) is denoted by the pair $\sigma = (\mathcal{X}, \mathcal{C})$, where:

- $\mathcal{X} = \{x_1, \ldots, x_n\}$ is the set of *propositional variables* (*events* in the CPA context);
- $\mathcal{C} = \{C_1, \ldots, C_m\}$ is a set of *clauses*.

A *truth assignment* or, equivalently, a *possible world*, is an assignment of values $\{\mathit{true}, \mathit{false}\}$ to the variables in $\mathcal{X}$, or equivalently, a vector $x \in \{0,1\}^n$. Notice that the number of possible worlds is $N = 2^n$. We denote by $X = \{x^{(1)}, x^{(2)}, \ldots, x^{(N)}\}$ the set of possible worlds; W is the $n \times N$ matrix with columns $x^{(1)}, x^{(2)}, \ldots, x^{(N)}$.

A truth assignment defines a corresponding *truth vector* $y \in \{0,1\}^m$, where for $1 \leq i \leq m$ it is $y_i = 1$ if clause C_i is satisfied and $y_i = 0$ otherwise; y_i is the *truth value* of clause C_i. We denote by $Y = \{y^{(1)}, y^{(2)}, \ldots, y^{(N)}\}$ the set of N truth vectors corresponding to the N possible worlds; note that some of the truth vectors may be identical. Denote by M the $m \times N$ matrix with columns $y^{(1)}, y^{(2)}, \ldots, y^{(N)}$.

A *probability assignment* to the possible worlds is a nonnegative N-vector p whose entries sum to one. Given a probability assignment p to the possible worlds:

- p gives a convex combination of the vertices of the unit hypercube in $\mathbb{R}^n$, i.e. it defines a vector $r(p) = Wp$ in the hypercube. The value $r(p)_j$ will be interpreted as the probability of variable x_j under the probability assignment p.
- the probability π_i of a clause C_i is the sum of the probabilities of the possible worlds in which the clause is true, i.e. $\pi_i = (Mp)_i$.

Suppose that $\pi \in [0,1]^m$ is a given assignment of probabilities to the clauses in σ. Problem PSAT is to determine if π is *consistent*, i.e. there exists a probability assignment p such that $\pi = Mp$. Therefore, PSAT can be formulated as a linear programming problem:

$$P_{\text{PSAT}} = \begin{cases} \min \mathbf{0}p \\ \text{s.t.} \\ Mp \;=\; \pi \\ \mathbf{1}p \;=\; 1 \\ p \;\geq\; 0 \end{cases}$$

where π is consistent if and only if the optimal value is zero.

Suppose now that $\eta \in [0,1]^n$ is an assessment of probabilities to the variables in σ. Problem CPA is to determine if η is *coherent*, that is, there exists a probability assignment p such that $Wp = \eta$, with the further constraint that a nonzero probability can be assigned only to a truth assignment satisfying the formula σ. Thus CPA can be reduced to a PSAT instance obtained from σ as follows:

- add a positive unit clauses U_j for each variable $x_j \in \mathcal{X}$;

- each "original" clause $C_i \in \mathcal{C}$ is assigned probability 1;

- each unit clause U_j is assigned probability η_j.

However, a simpler formulation is possible. Let us denote by $X^S \subseteq X$ the set of truth assignments satisfying the formula σ, and by W^S the submatrix of W whose columns are the vectors in X^S; accordingly, denote by p^S a probability assignment to the possible worlds in X^S. Then, η is coherent if and only if $\eta = W^S p^S$ for some p^S. This gives the LP:

$$P_{\text{CPA}} = \begin{cases} \min \mathbf{0} p^S & \\ \text{s.t.} & \\ W^S p^S & = \ \eta \\ \quad \mathbf{1} p^S & = \ 1 \\ \quad\ \ p^S & \geq \ 0 \end{cases}$$

Effective *column generation* techniques have been developed for problem P_{PSAT}, see [8] for the details omitted here. At each step of the column generation procedure, we have to solve the following *pricing problem*: we are given a vector $u \in \mathbb{R}^m$ of dual variables, and we have to find the vector $y^{(i)} \in Y$ that maximizes the product $u^T y$ over Y, that is

$$y^{(i)} = \arg \max_{y \in Y} u^T y.$$

Clearly, the pricing problem is equivalent to finding the truth assignment $x^{(i)}$ that maximizes the sum of the weigths of the satisfied clauses, where the weight of clause C_i is u_i. In other words, the pricing problems is a *weighted maximum satisfiability* problem (Weighted Max SAT).

A similar technique can be applied to problem P_{CPA}. In this case, the pricing problem is to find the vector $x^{(i)} \in X^S$ that maximizes the product $u^T x$ over X^S, that is

$$x^{(i)} = \arg \max_{x \in X^S} u^T x.$$

This amounts to find the truth assignment satisfying σ that maximizes the sum of the weights of the variables set to one, where the weight of variable x_j is u_j. By straightforward techniques, this problem can be transformed into a minimization problem usually referred to as the (weighted) *minimum satisfiability* problem (Weighted Min SAT).

It is known (see [8]) that PSAT is polynomially solvable for a class of formulas if Max SAT is polynomially solvable on that class. Actually, a stronger relation holds (see e.g. [10]): on a class of formulas, PSAT (CPA) can be solved polynomially *if and only if* Weighted Max SAT (resp. Weighted Min SAT) can be solved polynomially. This relation provides helpful guidelines for research on easy cases of PSAT and CPA, and suggests some observations on the result described in the next section.

3 Ideal CPA Instances

Here we shall adapt to CPA the *extended set covering* formulation proposed for PSAT in [1]. Based on this formulation, a linear time algorithm for PSAT instances *represented by hypertrees* was developed in that paper. Of course, the same algorithm can be applied to CPA on the same class of formulas. However, if CPA is considered, a stronger result applies to a broader class of formulas. For this class, CPA reduces to verify that the probability assessment satisfies a compact system of linear inequalities.

Given the formula $\sigma = (\mathcal{X}, \mathcal{C})$, consider a vector of binary variables $x \in \{0,1\}^n$. It is well known that σ is satisfiable if and only if the following system has a solution:

$$Ax \;\geq\; \mathbf{1} - n(A); \tag{1}$$

where A is the clause-variable incidence matrix of σ, and the vector $n(A)$ gives the number of negative entries in each row of A. In other words, the set X^S of truth assignments satisfying σ can be described as the set of vectors $x \in \{0,1\}^n$ satisfying the constraints (1):

$$X^S = \Big\{ x \in \{0,1\}^n : \; Ax \geq \mathbf{1} - n(A) \Big\}.$$

As discussed before, a probability assessment η is coherent if and only if it is a convex combination of vectors in X^S, that is, if η belongs to the *convex hull* of X^S, denoted by $\mathrm{conv}(X^S)$.

Consider now the solutions of the *linear relaxation* of constraints (1), i.e. the set:

$$\overline{X} = \Big\{ x \in [0,1]^n : \; Ax \geq \mathbf{1} - n(A) \Big\}.$$

Set $\overline{X}$ is the *generalized set covering polytope* associated with matrix A. The polytope $\overline{X}$ is *integral* if its vertices are integer vectors, that is, if $\overline{X} = \mathrm{conv}(X^S)$. In this case, we say that matrix A is *ideal*. Obviously, if $\overline{X}$ is integral, CPA can be reduced to checking whether $\eta \in \overline{X}$, that is, to check whether η is a solution of the system (1). Clearly, this can be done in $O(L)$ time, where L is the *length* (total number of literal occurrencies) of σ.

Let us say that a formula σ is *ideal* if the corresponding matrix A is ideal. We have the following:

Result 1 *CPA for ideal formulas can be solved in linear time.*

It must be remarked that this result does not extend to PSAT, where the set of solutions is represented by a more complex system of inequalities, that can define a non-integral polytope even if A is ideal.

The *balanced* matrices are a relevant sub-class of ideal matrices. Logic problems for formulas that give rise to balanced matrices have been investigated in [6]. Interestingly, both SAT and CPA can be easily solved in linear time for these formulas, while a similar result (a simple "combinatorial" method) is not known for Max SAT and Min SAT; however, these problems reduce to linear programs, where A is the constraint matrix. In light of the relation between Min SAT and CPA pointed out before, we might raise the question of whether a simple algorithm for Min SAT can be devised.

4 CPA and Horn Formulae with Uncertainty

Horn Formulae with uncertainty [2] (HFU for short) are a computationally simple tool for dealing with *uncertainty* in knowledge bases. In practice, a HFU corresponds to a propositional rule-based system, where *facts* are represented by atomic propositions and *rules* are represented by Horn clauses. For each proposition x_i, there exists a *certainty factor* $\alpha_i \in [0,1]$ representing the "degree of belief" that fact x_i is true. Some of these factors are fixed, and correspond to evidence provided from outside. In addition, a Horn clause:

$$x_0 \longleftarrow x_1 \wedge \ldots \wedge x_k$$

implies that fact x_0 has a certainty factor *at least* α, where:

$$\alpha = \min_{1 \leq i \leq k} \alpha_i.$$

Thus α_0 is not less than the maximum factor α obtained from the clauses with consequence x_0. As noted in [2], this is coherent with the definition of logical connectives in *fuzzy logic*.

A *minimum model* of an HFU assigns to each fact x_i the *minimum* factor α_i consistent with the above rules. A minimum model can be find in linear time w.r.t. the total size of the Horn clauses. Moreover, HFUs are suitable for a dynamic environment, where new rules, new facts, as well as new evidence supporting known facts can be added on–line. Indeed, efficient incremental algorithms are presented in [2]. Remark that these algorithms are based on a *directed hypergraph* representation; in fact, finding the minimum model reduces to solving a *shortest hyperpath* problem.

In general, a HFU may include facts whose certainty factor must not exceed a given threshold: if this happen, an inconsistency is detected. In [2], a single fact (corresponding to the proposition *false*) with a zero threshold is included.

Assume that $\sigma = (\mathcal{X}, \mathcal{C})$ is the Horn formula defined by the facts and rules in a HFU. The certainty factors α_i can be considered as a probability assessment on $\mathcal{X}$. The following fact holds true:

Result 2 *a minimum model defines a coherent probability assessment*

Note that the converse is not true, that is, a coherent probability assessment on σ does not necessarily provide a consistent model for an HFU. In other words, HFU are a rather restrictive system, where efficiency is obtained at the expenses of generality.

Result 2 can be proved constructively by a simple algorithm that builds a probability assignment to possible worlds where σ is satisfied. Let η be the probability assessment defined by the HFU model; the algorithm can be described as follows:

Step 0 let $S = \mathcal{X}$; $R = 0$; $\delta = \eta$;

Step 1 remove from S each proposition x_i such that $\delta_i = 0$;
if S is empty, go to Step 4;

Step 2 assign probability

$$\rho = \min_{x_i \in S} \delta_i$$

to the possible world where $x_i = \mathit{true}$ if $x_i \in S$ and $x_i = \mathit{false}$ otherwise;

Step 3 let $R = R + \rho$; for each $x_i \in S$, let $\delta_i = \delta_i - \rho$; go to Step 1;

Step 4 if $R < 1$ assign probability $1 - R$ to the possible world where $x_i = \mathit{false}$ for each $x_i \in \mathcal{X}$.

It is easy to verify that at most $n+1$ truth assignments are considered, and all of them satisfy σ; moreover, the sum of the assigned probabilities is one, and the resulting probability assessment to the propositions is exactly η.

Result 2 shows that the models provided by HFUs are consistent in probabilistic logic; a semantic interpretation can be provided considering the CPA problem. An interesting direction for further research is to find an extension of HFUs that retains the same computational efficiency, and where uncertainty is associated with both facts and rules; in this case, a semantic model could be based on the PSAT problem.

5 Open Questions

So far, we have seen examples where CPA is "easier" than PSAT. Similar examples are known for Min SAT and Max SAT: for Horn formulas and nonnegative weights, Min SAT takes linear time and Max SAT is NP-hard. This suggests the following questions:

1. can we establish a hierarchy between CPA and PSAT?

2. can we extend the hierarchy to the separation (pricing) problems, Min SAT and Max SAT?

A related question is the following. As shown above, CPA can be easily reduced to PSAT by adding unit clauses; in a similar way, Min SAT can be easily reduced to Max SAT. It is also easy to reduce PSAT to a variant of CPA, where the probabilities of variables $\mathcal{X}$ are left unspecified. Accordingly, Max SAT can be easily reduced to a Min SAT problem where variables $\mathcal{X}$ have zero weight. Actually, since CPA and

PSAT are NP-complete problems, PSAT can be reduced to the "true" CPA; however, a "natural" polynomial time reduction is not known. Note that such a reduction should somehow "capture", in terms of logical constraints, the relations between variables and clauses that are "hidden" in the PSAT model. On the other hand, it may be the case that such a reduction does not exist. In this case, we might conclude that the PSAT model is more expressive than CPA, that is, assigning probabilities to logical relations between events is more powerful than assigning probabilities to logically related events.

References

[1] K. A. Andersen and D. Pretolani, "Easy Cases of Probabilistic Satisfiability", *Annals of Mathematics and A.I.* **33** No. 1 (2001) 69–91.

[2] G. Ausiello and R. Giaccio, "On-line algorithms for satisfiability formulae with uncertainty", *Theoretical Computer Science* **171** (1997) 3–24.

[3] M. Baioletti, A. Capotorti, S. Tulipani, B. Vantaggi, "Simplification Rules for the Coherent Probability Assessment Problem", Dipartimento di Matematica e Informatica, Università di Perugia, Italy (2001).

[4] G. Boole, *An investigation of the laws of thought*, Walton and Maberley, London (1854).

[5] G. Coletti, "Coherence Principles for Handling Qualitative and Quantitative Partial Probabilistic Assessment", *Mathware & Soft Computing* **3** (1996) 159–172.

[6] M. Conforti and G. Cornuéjols, "A class of logic problems solvable by Linear Programming", *Journal of the Association for Computing Machinery* **42** (1995) 1107–1113.

[7] B. de Finetti, *Teoria della probabilità*, vol. I,II, Einaudi, Torino (1970). Eng. transl. *Theory of Probability*, vol. I,II, Wiley & Sons (1974).

[8] G. Georgakopoulos, D. Kavvadias and C. H. Papadimitriou, "Probabilistic Satisfiability", *Journal of Complexity* **4** (1988) 1–11.

[9] T. Hailperin, *Boole's Logic and Probability*, 2nd ed., Studies in Logic and the Foundations of Mathematics vol. 85, North-Holland, Amsterdam (1986).

[10] G. L. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Optimization*, John Wiley & sons (1988).

[11] N. J. Nilsson, "Probabilistic Logic", *Artificial Intelligence* **28** (1986) 71–87.