GEO-SAT: A New Approach for Knowledge-Based Agent Decision Making

Thomas C. Henderson, Amelia Lessen, Ishaan Rajan, Tessa Nishida, Kutay Eken, Xiuyi Fan, David Sacharny, Amar Mitiche, and Thatcher Geary

University of Utah, Salt Lake City UT 84112, USA,
tch@cs.utah.edu,
WWW home page: www.cs.utah.edu/~tch

Abstract. Logical agents base their action selection decisions on inferences made over a logical knowledge base. Given a propositional logic knowledge base expressed in Conjunctive Normal Form (CNF), the knowledge can be converted into a geometrical format, and subsequent analysis takes place as geometrical operations on the feasible region in that representation. Two geometric representations are presented: the n-dimensional hypercube in Euclidean geometry and the n-dimensional Poincaré disk in non-Euclidean geometry. Based on these representations, two novel methods are proposed to: (1) find SAT solutions for the knowledge base (i.e., a truth assignment to each logical variable which makes the CNF sentence true), and (2) find a reasonable approximation to the atom probabilities given the current set of information. This allows agents to determine the semantics (truth) of the world as well as to estimate the probability of truth. The geometric method provides an efficient heuristic approach to solving SAT for CNF knowledge bases, and provides polynomial-time solutions of probabilistic SAT for independent variables, and good PSAT estimates for non-independent logical variables.

1 Introduction and Background

Given a propositional calculus knowledge base represented in Conjunctive Normal Form (CNF), an agent can find out information about the world by finding new sentences that are entailed by the current knowledge. In this way, an agent can determine proper actions. To do so, it may be useful to determine whether the CNF sentence has a solution (satisfiable) or not (unsatisfiable). This is called the SAT problem (for SATisfiability) problem. The SAT problem is NP complete [21].

Another aspect of interest to the agent is the probability that a specific atom (logical variable) is true. For example, for the CNF $S = A \vee B$, there are three solutions, (0, 1), (1, 0), (1, 1), which satisfy S and, assuming equal likelihood for all solutions, the probability of A is 2/3, and the probability of B is 2/3 Note that this is the mean of the models (truth assignments to variables) which satisfy the sentence. One way to determine the atom probabilities is to solve the *Probabilistic SAT* (PSAT) problem when each conjunct, C_i, is given a probability, p_i [9,17]. That is, given n logical variables, there are 2^n unique truth assignments (also called models or the complete conjunction set) to the variables. The set of all models is called Ω, and ω_i is the model with binary assignments corresponding to the binary representation of $i-1$; e.g., ω_1 is all zero assignments - all false. The PSAT problem consists of determining a probability distribution, $\pi : \Omega \rightarrow [0,1]$ such that $\sum_{i=1}^{2^n} \pi(\omega_i) = 1$, and $\sum_{\omega_i \models C_j} \pi(\omega_i) = p_j$, for all conjunct probabilities, p_i. The probability of an atom is then found as $Prob(A) = \sum_{\omega_i \models A} \pi(\omega_i)$. We have also previously described how to solve the Probabilistic Sentence Satisfiability Problem (PSSAT) [12] which in certain cases provides a PSAT solution (i.e., given independent variables). The methods we are proposing differ from standard methods in that we solve linear or nonlinear systems of equations rather than having to consider the full joint probability distribution over the variables (e.g., like Bayesian networks [18] or Markov Logic Networks [7]).

The purpose of this study is to investigate *Chop-SAT* as an alternative way to answer these questions about SAT and PSAT [13,15]. Work on the use of cutting planes started with Gomory [11] who sought integer solutions for linear programs. Given the semantics of the literals in a disjunction, then a linear inequality can be formed summing x_i for atoms in the clause and $(1 - x_i)$ for negated atoms in the clause and setting this to be greater than or equal to 1. Next, a $\{0,1\}$ solution is sought resulting in an integer linear programming problem. If a non-$\{0,1\}$ solution is found, Gomory proposed a way to separate (via a *cutting plane*) that solution from all integer solutions. This method has been used in finding lower complexity ways to provide theorems for proving the boundedness of polytopes, cutting plane proofs for unsatisfiable sentences, pseudo-Boolean optimization, etc. (see [2–6]). The *Chop-SAT* method was discovered independently and is based on a different set of insights into the nature of the CNF form.

Based on the *Chop-SAT* approach, the contributions here provide:

1. A method to determine whether a SAT solution exists, and
2. A method to determine an approximation to the atom probabilities.

2 Chop-SAT

A CNF sentence is the conjunction of a set of disjunctions where each disjunction is a literal (i.e., either an atom or the negation of an atom). A CNF sentence is then represented as $S = C_1 \wedge C_2 \wedge \ldots \wedge C_m$, where $C_i = L_{i,1} \vee L_{i,2} \vee \ldots \vee L_{i,k_i}$ where $L_{i,j} = a_p$ or $L_{i,j} = \neg a_p$, and a_p is an atom.

The satisfiability of a CNF sentence, S, over n variables can be converted to a geometric problem as follows. Consider the hypercube of dimension n centered at $[\frac{1}{2}, \frac{1}{2}, \ldots, \frac{1}{2}]$; call it H_n. Then each vertex of H_n represents a model for n variables. The vertexes also represent assignments of probability 0 or 1 for the truth of each variable. Every other point in H_n can be considered to give a probability on the interval $[0,1]$ for each variable. E.g., the center of H_n represents a probability assignment of 1/2 for each atom.