Random verification of Knowledge-Based Systems with uncertainty

Michel de Rougemont
ENSTA [1]
32 Blvd. Victor, F-75015 Paris, France
&
Laboratoire de Recherche en Informatique,
Université de Paris-Sud,
F-91405 Orsay, France
e-mail: mdr@lri.fr

Abstract: We describe a new formal method to verify knowledge-based systems dealing with uncertain data following a probabilistic model. For a large class of uncertainty models, it is a hard combinatorial problem ($PSPACE$) to verify that the probability of correctness is greater than k. We present a method, based on the interactive proof systems, that allows a random verifier to check a prover in $O(n^2)$. We then show how to use it in practice to rate Knowledge-Based systems.

1 Introduction

A large class of Knowledge-Based (KB) systems can be viewed as systems taking as inputs a finite structure U, and a set of logical rules in the language of U. Assume we are interested in a query Q defined in such a logical setting. Without loss of generality, suppose we have a finite graph $G = (D_n, E)$ where E is a binary edge relation on D_n, a finite domain with n elements. Assume also that we have an uncertainty measure, defined as an arbitrary function p which assigns for every edge e, a rational number $p(e) \in [0,1]$ representing the probability that the edge e exists (does not fail). The probabilistic function p defines a probabilistic measure on the subgraphs $G' \subseteq G$ and the problem is to compute :

$$Pr\{ [Q]^G = [Q]^{G'} \}$$

We want to know the robustness of our KB system, i.e. the probability that the answer to the query in an uncertain graph (G') coincides with the answer to the query without uncertainty. If Q is the classical GAP[2] query, this problem is called *Graph Reliability*, and is a well understood combinatorial problem [Val79]. We want to verify the following problem :

$$\textit{Given a query } Q \textit{ and a rational } k \in [0,1], \textit{ is } Pr\{ [Q]^G = [Q]^{G'} \} > k \quad ?$$

It is a canonical problem, that would allow to compare two KB-systems, KB_1 and KB_2. Which one is more reliable for a query Q? This is a central question for Knowledge-Bases dealing with uncertainty, and it is a hard combinatorial problem. (In fact $\#P$-complete in the case of GAP).

We present a *random and interactive* solution to this problem, where a verifier checks a prover claiming to solve this problem. This idea is used to test the robustness of planning in robotics [dRDf92] and to obtain an efficient method to compare strategies controling mobile robots. The robot does not claim to solve a $\#P$-complete problem, but claims to solve GAP on a class K of graphs of size $n < 1000$ with a simple algorithm. In the protocol, the verifier queries the prover on smaller graphs of the same class K, and hence can convince himself quickly of the validity of the robot's claim. In general it is a method to study problems with uncertainty, and to test the realizability of various hypotheses.

We sketch such constructions, and show its use in practice.

[1] Ecole Nationale Supérieure de Techniques Avancées.

[2] Given two points s, t, decide if there is a path from s to t.

2 Proof verification

The notion of an *interactive proof system* or *interactive protocol* was introduced by Goldwasser, Micali and Rackoff [SSC89] and independently by Babai [Bab85, Bab88]. Intuitively, it is a way by which an infinitely powerful prover can convince using interaction a polynomially powerful probabilistic verifier about the membership of elements in some language, but only for those elements, which are indeed in the language. More formally, a language L belongs to the class IP, if there is a probabilistic polynomial time verifier $\mathcal{V}$, and a prover $\mathcal{P}$ such that for every $x \in L$, $\mathcal{P}$ can convince $\mathcal{V}$ to accept x with overwhelming probability, but for every $x \notin L$, no prover $\mathcal{P}'$ can convince $\mathcal{V}$ to accept x with more than negligible probability.

The idea of an Interactive protocol is to check a prover on limited instances, and not exhaustively. Instead of checking GAP on all subgraphs G' of G (exponentially many), we will run the protocol on $O(n^2)$ subgraphs.

3 An interactive protocol for *GAP*

It is based on the following decomposition presented in the case where we deal with valued graphs (there is an integer associated with every edge) :

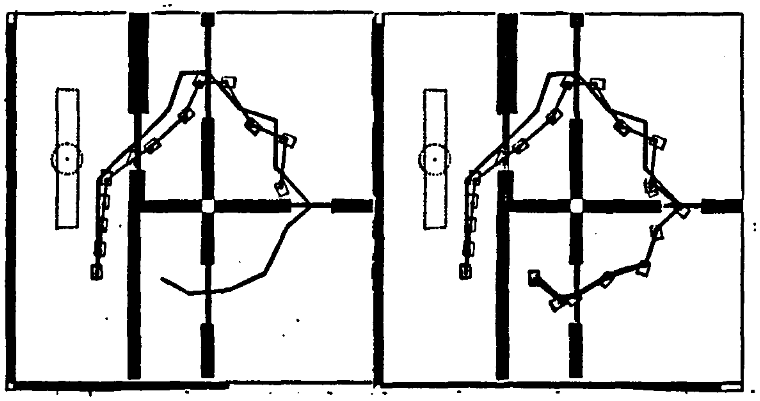
1. $\mathcal{V}$ asks $\mathcal{P}$ for k, the probability of GAP.
2. $\mathcal{V}$ generates a *particular* subgraph G_i, with random values on the edges, and asks $\mathcal{P}$ the same question.
3. $\mathcal{V}$ combines $\mathcal{P}$'s answers with the previous one and runs a simple test. Then iterate.

With at most $O(n^2)$ operations, the verifier checks the prover up to some ϵ. This technique generalizes to many other queries.

4 Applications to KB-systems

An application of these techniques has been used in Robotics to test controling strategies (using sensors and standard planning) in an uncertainty model [dRDf92]. The figure below shows a simulator testing a KB-system controling a mobile robot : the uncertainty is computed by a random generator and the control system is then tested on many experiments.

The scene below shows obstacles in black and the robot has to traverse 4 doors from a starting position s to a final position t, avoiding potential unknown obstacles. The left image shows an unsuccessful experiment : the robot blocks just before the third door. The right image shows a successful experiment, where the robot reaches t. Each image shows a virtual path (no uncertainty), and a real path (with uncertainty). If we run many experiments, we can estimate the probability of success by just counting the number of successfull experiments. We get some valuable information, but how many experiments should we run?



The *IP* protocol is equivalent to the *statistical evidence* given by the simulator. It answers the fundamental question : how many tests are necessary to check a property? In the case of GAP, we need $O(n^2)$ interactions, and some other queries may need less.

The protocol is used to evaluate the reliability of a KB system and to compare the reliability of two different Knowledge Bases. Starting with a given system, we were able to refine it steps by steps, with an improved reliability. It is then implemented on the real mobile robot, and behaves as predicted in a random uncertain environment. We will show how to generalize this approach to arbitrary queries expressed in some logic-based language (Datalog and its variations). We will also discuss other uncertainty models, for which we do not know the corresponding *IP* protocols.

References

[Bab85] L. Babai. Trading group theory for randomness. *Symposium on the Theory of Computing,* pages 421–429, 1985.

[Bab88] L. Babai. E-mail and the unexpected power of interaction. *Structure in Complexity theory,* 1988.

[dRDf92] Michel de Rougemont and Juan F. Diaz-frias. A theory of robust planning. In *Proceedings of the I.E.E.E. International Conference on Robotics and Automation,* 1992.

[SSC89] Goldwasser S., Micali S., and Rackoff C. The knowledge complexity of interactive proof systems. *SIAM Journal of Computing,* pages 186–208, 1989.

[Val79] L. Valiant. The complexity of enumeration and reliability problems. *SIAM,* 8(3), 1979.