Sampling and Soundness: Can We Have Both?

Carla Gomes[1], Jörg Hoffmann[2], Ashish Sabharwal[1], and Bart Selman[1]

[1] Cornell University, Ithaca, NY, USA, gomes|sabhar|selman@cs.cornell.edu
[2] University of Innsbruck, DERI Institute, Austria, joerg.hoffmann@deri.at

Abstract. Recent research on model counting in CNF formulas has shown that a certain sampling method can yield results that are sound with a provably high probability. The key idea is to iteratively restrict the search space, and to randomly choose which part to consider. The expected value of this sampled count is equal to the real count. If one minimizes over several trials, and purposefully underestimates the outcome of each trial by a constant factor, then the probability that the sampled count exceeds the real count decreases exponentially in the number of trials. This method has proven to be quite successful for many CNF formulas. The big question is: Can we devise similar methods for reasoning in the Semantic Web? Is it possible to obtain provably high-quality results based on sampling?

1 Sampling and Model Counting

Recent research on model counting in propositional CNFs [2, 1] has shown that a certain sampling method can yield results that are sound with a provably high probability. The method proceeds thus: s times, iteratively throw a constraint cutting the space of potential models in half; uniformly choose one of the halfs to continue with; count the remaining models exactly once that is feasible, and multiply the result with 2^s.

Here, the "constraints" can be as simple as fixing the value of some variables to true or false. The constraints can also be more complicated, e.g., requiring the values of two variables to be the same, or requiring them to be different. In principle, any constraint is possible as long as the number of value assignments that satisfy the constraint (in isolation) is equal to the number of value assignments that don't. One chooses the "half to consider" by throwing either the constraint itself, or its negation, adding that conjunctively to the CNF formula. Once enough constraints are added, counting the models of the formula exactly is easy.[1] Since the half to be considered is chosen randomly, the expected value of the obtained count is equal to the real count.

One execution of the above method is one "trial" to count the models. Of course, the outcome of each trial may be wrong. However, with a simple trick one can ensure that the likelihood of over-estimating the true count decreases rapidly. Instead of multiplying the number of remaining models with 2^s, multiply only with $2^{s-\alpha}$, where α is fixed. Minimize the outcome over t trials. Then, the probability that the minimum count exceeds the real count is less than $2^{-\alpha t}$. This can be proved exploiting Markov's inequality, stating that, for any k and for any random variable X, $Pr[X > kE[X]] < 1/k$. The outcome of the count in any single trial plays the role of X; therefore, any single trial exceeds the real count with a probability less than $2^{-\alpha}$, from which the claim follows.

[1] This depends on the kind of constraint; it holds, e.g., if variable values are being fixed.

The confidence in the lower bound holds irrespectively of how the constraints are chosen. But of course, this choice is important: if the constraint does not cut the *real* models, i.e., the models of the overall formula, in half, then the sampled count is erroneous. So one should heuristically choose constraints that are likely to cut the real models in about half. If bad choices are made, then the variance of the outcome will be high, and the lower bound will be overly generous. It has been shown that very good lower bounds can be quickly obtained in many hard CNF formulas, if constraints are chosen based on heuristic information gathered with local search techniques.

2 Application to the Semantic Web

Can we devise similar methods for reasoning in the Semantic Web? Is it possible to obtain provably high-quality outcomes based on sampling? We cannot even try to answer this question comprehensively. We list some things that spring to mind.

If the task to be performed involves counting, or can be formulated as such, then it seems pretty clear that the sampling method can be adapted. For example, say one has an RDF database, and wants to count how many triples comply with a given query. In such situations, the main conceptual issue that needs to be clarified is if/how the notion of “constraints” can be adapted: How to cut the set of all (potential) RDF triples in half? Apart from that, severe technical challenges may have to be overcome as to how to “throw” a constraint, and how to efficiently count the “remaining models”.

It is a priori less clear if/how situations of a “yes/no” nature can be tackled. What if we want to check whether some logical statement ϕ follows from a huge database of facts and axioms? A straightforward adaptation would throw “constraints” as additional axioms or facts. Of course, this raises a big issue regarding how to make sure that the additional constraints actually make the reasoning more efficient. If the implication does not hold in the enriched database, we know that it does not hold in the original one. Otherwise, potentially one can upper-bound the probability that the implication does not hold, by exploiting the correspondence between proving an implication, proving unsatisfiability, and proving upper bounds on the number of models. For that purpose, the sampling method would first need to be adapted to derive upper bounds instead.

An alternative adaptation would be to instead throw “constraints” removing parts of the database, and check whether the remaining parts imply ϕ. But can we draw any conclusions from that? In the general case, no: the constraints might be of a form so that those parts of the database responsible for implying ϕ are on different sides of the constraint. Even if that never happens, the probability of retaining the responsible parts decreases exponentially with the number of constraints thrown. Are there interesting special cases where these difficulties do not appear, or at least do not matter in practice?

References

1. Carla Gomes, Jörg Hoffmann, Ashish Sabharwal, and Bart Selman. From sampling to model counting. In *Proc. IJCAI'07*, pages 2293–2299, 2007.
2. Carla Gomes, Ashish Sabharwal, and Bart Selman. Model counting: A new strategy for obtaining good bounds. In *Proc. AAAI'07*, pages 54–61, 2007.