Theoretical Computer Science 649 (2016) 1–24

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Command-based importance sampling for statistical model checking

Cyrille Jegourel [a,*], Axel Legay [b], Sean Sedwards [b]

[a] National University of Singapore, Singapore
[b] Inria Rennes – Bretagne Atlantique, France

ARTICLE INFO

Article history:
Received 19 January 2015
Received in revised form 14 July 2016
Accepted 4 August 2016
Available online 23 August 2016
Communicated by J.-F. Raskin

Keywords:
Statistical model checking
Monte Carlo
Rare events
Importance sampling
Cross entropy
Guarded commands

ABSTRACT

Statistical model checking avoids the exponential growth of states of numerical model checking, but rare properties are costly to verify. Importance sampling can reduce the cost if good importance sampling distributions can be found efficiently.

Our approach uses a tractable cross-entropy minimisation algorithm to find an optimal parametrised importance sampling distribution. In contrast to previous work, our algorithm uses a naturally defined low dimensional vector to specify the distribution, thus avoiding an explicit representation of a transition matrix. Our parametrisation leads to a unique optimum and is shown to produce many orders of magnitude improvement in efficiency on various models. In this work we link the existence of optimal importance sampling distributions to logical properties and show how our parametrisation affects this link. We also motivate and present simple algorithms to create the initial distribution necessary for cross-entropy minimisation. Finally, we discuss the open challenge of defining error bounds with importance sampling and describe how our optimal parametrised distributions may be used to infer qualitative confidence.

1. Introduction

The most common method to ensure the correctness of a system is by testing it with a number of test cases having predicted outcomes that can highlight specific problems. Such testing techniques remain the default in industrial contexts and have also been incorporated into sophisticated tools [1]. Despite this, testing is limited by the need to hypothesise scenarios that may cause failure and the fact that a reasonable set of test cases is unlikely to cover all possible eventualities. Errors and modes of failure in complex systems may remain undetected and quantifying the likelihood of failure using a series of test cases is difficult.

Static analysis has been successful in debugging very large systems [2], but its ability to analyse dynamical properties is limited by its level of abstraction. In contrast, model checking is a fine-grained exhaustive technique that verifies whether a system satisfies a dynamical temporal logic property under all possible scenarios. In recognition of the existence of non-deterministic and probabilistic systems, and the fact that a Boolean answer is not always useful, *numerical* model checking quantifies the probability that a system satisfies a property. Numerical model checking offers precise and accurate analysis by exhaustively exploring the state space of probabilistic systems. The result of this technique is the notionally exact probability (i.e., within the limits of numerical precision and convergence stability) that a system will satisfy a property

* Corresponding author.
E-mail addresses: jegourel@comp.nus.edu.sg (C. Jegourel), axel.legay@inria.fr (A. Legay), sean.sedwards@inria.fr (S. Sedwards).

http://dx.doi.org/10.1016/j.tcs.2016.08.009
0304-3975/

of interest, however the exponential growth of the state space limits its applicability. The typical state limit of exhaustive approaches usually represents an insignificant fraction of the state space of "real" systems. Such systems may have tens of orders of magnitude more states than the number of protons in the universe ($\approx 10^{80}$).

Symbolic model checking using efficient data structures can make certain very large models tractable [3]. It may also be possible to construct simpler but behaviourally equivalent abstractions using various symmetry reduction techniques, such as partial order reduction, bisimulation and lumping [4]. Compositional approaches may also help. In particular, components of a system may be specified in such a way that each is tractable to analysis, while their properties guarantee that certain faults are impossible. Despite these techniques, however, the size, unpredictability and heterogeneity of real systems [5] often make numerical techniques infeasible. Moreover, even if a system has been specified not to misbehave, it is nevertheless necessary to check that it meets its specification.

While the 'state explosion problem' [6] is unlikely to ever be entirely solved for all systems, simulation-based approaches are becoming increasingly tractable due to the availability of high performance parallel hardware and algorithms. In particular, *statistical* model checking (SMC) combines the simplicity of testing with the formality of numerical model checking. The core idea of SMC is to create multiple independent execution traces of the system and individually verify whether they satisfy some formally specified property. The proportion of satisfying traces is an estimate of the probability that the system satisfies the property. By thus modelling the executions of a system as a Bernoulli random variable, the absolute error of the estimate can be bounded using, for example, a confidence interval [7, Chap. 1] or a Chernoff bound [8–10]. It is also possible to use efficient techniques, such as Bayesian inference [11] and hypothesis testing [12,13], to decide with specified statistical confidence whether the probability of a property is above or below a given threshold.

Knowing a result with less than 100% confidence is often sufficient in real applications, since the confidence bounds may be made arbitrarily tight. Moreover, a swiftly achieved approximation may prevent a lot of wasted time during model design. For many complex systems, SMC offers the only feasible means of quantifying performance. Evidence of this is that SMC has been used to find bugs in large, heterogeneous aircraft systems [5]. Dedicated SMC platforms include APMC [14], YMER [15], VESTA [16], PLASMA [17] and COSMOS [18]. Well-established numerical model checkers, such as PRISM [19] and UPPAAL [20], are now also including SMC engines. Indeed, since SMC may be applied to any discrete event trace obtained by stochastic simulation, [21] describes a modular library of SMC algorithms that may be used to construct domain-specific SMC tools.

SMC relies on multiple independent simulations, so it may be efficiently divided on parallel computer architectures, such as grids, clusters, clouds and general purpose computing on graphics processors (GPGPU). Despite this, rare properties require a challenging number of simulations. Standard error bounding strategies for SMC consider *absolute* error. As the probability of a property decreases, however, it is more useful to consider an error bound that is relative to the probability. The number of simulations required to bound the *relative* error, defined as the standard deviation of the estimate divided by its expectation, is inversely proportional to rarity. Hence, while SMC may make a verification task feasible, it may nevertheless be computationally intense. To address this problem, in this work we apply the variance reduction technique of *importance sampling* to statistical model checking.

Importance sampling works by simulating a system under a weighted (importance sampling) distribution that makes a property more likely to be seen. It then compensates the results by the weights, to estimate the probability under the original distribution. The concept arose from work on the 'Monte Carlo method' [22] in the 1940s and was originally used to quantify the performance of materials and solve otherwise intractable analytical problems with limited computer power (see, e.g., [23]).

For importance sampling to be effective it is necessary to define a "good" importance sampling distribution: (*i*) the property of interest must be seen frequently in simulations and (*ii*) the distribution of the simulation traces that satisfy the property in the importance sampling distribution must be as close as possible to the normalised distribution of the same traces in the original distribution. Failure to consider both (*i*) and (*ii*) can result in gross errors and overestimates of confidence. Moreover, the process of finding a good importance sampling distribution must itself be efficient and, in particular, should not rely on iterating over all the states or transitions of the system. The algorithms we present in this work address all these issues.

The term 'rare event' is ubiquitous in the literature. Here we specifically consider rare *properties* of paths, defined in bounded temporal logic (bounded by time or number of steps). This extends the common notion of rarity from states to paths. States are rare if the probability of reaching them from the initial state is small. Paths are rare if the probability of executing their sequence of states is unlikely–whether or not the states themselves are rare. Rare properties are therefore more general than rare states, however the distinction does not significantly alter the mathematical derivation of our algorithms. It can nevertheless affect the existence of the so-called "zero variance" optimal importance sampling distribution as a simple re-parametrisation of the states and transitions of the original system. We explore this important subject in Section 7.

1.1. Contribution

This work extends [24], describing the additional techniques necessary to apply our importance sampling framework for statistical model checking of rare events. We describe simple algorithms to initiate the cross-entropy minimisation process by finding at least a few traces that satisfy the property. We believe this subject has been glossed over in previous work.