Data-efficient Bayesian verification of parametric Markov chains

E. Polgreen[1], V.B. Wijesuriya[1], S. Haesaert[2], and A. Abate[1]

[1] Department of Computer Science, University of Oxford
[2] Department of Electrical Engineering, Eindhoven University of Technology

Abstract. Obtaining complete and accurate models for the formal verification of systems is often hard or impossible. We present a data-based verification approach that addresses incomplete model knowledge. We obtain experimental data from a system that can be modelled as a parametric Markov chain, and the desired system properties are expressed in a probabilistic logic. We propose a novel verification algorithm that quantifies the confidence that the underlying system satisfies a given property of interest, exploiting partial knowledge of the system to minimise the amount of sampled data required. Given a parameterised model of the system, the procedure first generates a feasible set of parameters corresponding to model instances satisfying a given probabilistic property. Simultaneously, we use Bayesian inference to obtain a probability distribution over the model parameter set from data sampled from the underlying system. The results of both steps are combined to compute a confidence that the underlying system satisfies the property. Our approach offers a framework to integrate Bayesian inference and formal verification, and in our experiments our new approach requires one order of magnitude less data than standard statistical model checking to achieve the same confidence.

1 Introduction

Complex engineering systems, such as autonomous vehicles and robots, are often safety-critical and demand high guarantees of correctness. Given a model of the system of interest, these guarantees can be obtained through formal methods, such as model checking [1]. However, the outcomes of these formal proofs are bound to models of the system of interest. Probabilistic logics such as PCTL [16] allow us to specify desired behaviours, which can be checked over probabilistic models. The uncertain stochastic dynamics of these models can be captured via parameterised Markov chains.

In this work we integrate the use of model checking techniques (for parameter synthesis over the model) with data-based approaches (for parametric Bayesian inference), in order to compute a confidence, based on observed data collected from the system, that a system satisfies a given specification. The proposed approach is distinctively different from statistical model checking (SMC) [31], a known data-based technique for model verification, and has a distinct set-up

and addresses a different objective: The original SMC algorithms target systems with fully known models that are too large for conventional model checking, and use the known models to generate simulated data; SMC has also been applied in a model-free setting where system-generated data is directly employed towards statistical validation of properties of interest; our technique instead targets partially known systems, captured as a parameterised model class that can be accessed, and still uses data collected from the original system.

In this work the parameterised model class is a parameterised Markov chain, and we propose a new model-based method to verify probabilistic properties over the system, which exploits this partial knowledge and computes a confidence guarantee via inference based on data generated by the underlying system. Whilst in general SMC requires a large amount of high-quality sample data covering the entire system behaviour to obtain good confidence results, we show that this method requires much less sample data, and that it can accommodate data with only partial coverage.

Our method can be elucidated in three phases. In the first phase, having a parameterised model of our partially known system, we use parameter synthesis to determine a set of feasible parameters over the given model class, namely those parameters corresponding to models of the system satisfying the given specification. Among a number of alternatives, we use an existing parameter synthesis method implemented in PRISM [19]. The second phase, which can be executed in parallel with the first, involves the use of Bayesian statistics to infer a distribution over the likely values of the parameters of the model class, given sample data collected from the underlying system. Finally, in the third phase we combine the outputs from the previous two phases to compute the confidence attached to the system satisfying the given specification.

Alongside the new methodology introduced in this work (first presented over different model class and properties in [12]), the key technical contribution resides in phase two: our algorithm introduces expansions of states and transitions of the parameterised Markov chain, which guarantees that the posterior probability distributions over the parameters can be obtained analytically, and can be integrated easily. The work discusses a case study, demonstrating the implementation of the algorithm, and a comparison with a standard SMC procedure. We argue that our approach requires smaller amounts of data than SMC to verify whether the system satisfies a given property up to a given confidence level; and further that our approach is more robust than SMC in situations where only limited data and a partial model is available.

Related Work Statistical Model Checking (SMC) [21] replaces numerical model-based procedures with empirical testing of formalised properties. SMC is limited to fully observable stochastic systems with little or no non-determinism, and may require the gathering a large set of measurements, or the generation of large numbers of sample trajectories from a complete system model. SMC techniques have been utilised to tackle verification of black box probabilistic systems [28], with no model of the system available, but this approach requires large amounts of data. Extensions towards the inclusion of non-determinism have been studied

in [17, 22], with preliminary steps towards Markov decision processes. Related to SMC techniques, [7, 24, 28] assume that the system is encompassed by a finite-state Markov chain and efficiently use data to learn and verify the corresponding model. Similarly, [2, 5] employ machine learning techniques to infer finite-state Markov models from data over given logical formulae.

Bayesian inference uses Bayes theorem to update the probability distribution of a set of hypotheses based on observed data [4]. Bayesian Inference for learning transition probabilities in Markov Processes is presented in [25], and the technique is widely used in planning in POMDP by [27] – in this work we do not consider models with actions or with the planning goals studied in this area.

2 Background

2.1 Parametrised Markov chains – syntax and semantics

Let S be a finite, non-empty set of states representing all possible configurations of the system being modelled. A discrete-time Markov chain (DTMC) is a stochastic time-homogeneous process over this set of states [1], as follows.

Definition 1 *A discrete-time Markov chain* $\mathbf{M}$ *is a tuple* $(S, \mathbb{T}, \iota_{init}, \mathrm{AP}, L)$*, where* S *is a finite, non-empty set of states,* $\mathbb{T} : S \times S \to [0, 1]$ *is the transition probability function such that for* $\forall s \in S : \sum_{s' \in S} \mathbb{T}(s, s') = 1$*. The function* $\iota_{init} : S \to [0, 1]$ *denotes an initial probability distribution over the states* S*, such that* $\sum_{s \in S} \iota_{init}(s) = 1$*. The states in* S *are labelled with atomic propositions* $a \in \mathrm{AP}$ *via the labelling function* $L : S \to 2^{\mathrm{AP}}$*.*

Consider the evolution of a Markov chain over a time horizon $t = 0, 1, \ldots, N_t$, with $N_t \in \mathbb{N}$. Then an execution of the process is characterised by a state trajectory given as $\{s_t | t = 0, 1, \ldots, N_t\}$. The transition function $\mathbb{T}(s, s')$ specifies for each state s the probability of moving to s' in one step, and hinges on the *Markov Property*, which states that the conditional probability distribution of the future possible states depends only on the current state, namely $\mathbb{P}(s' = s_{t+1} \mid s_t, ...s_0) = \mathbb{P}(s' = s_{t+1} \mid s_t)$. Furthermore, the definition of $\mathbf{M}$ requires that $\mathbb{T}$ is time homogeneous, that is $\mathbb{P}(s' = s_{t+1} \mid s_t = s) = \mathbb{P}(s' = s_t \mid s_{t-1} = s), \forall t \in \mathbb{N}$. The model is extended with (internal) non-determinism in order to express lack of complete knowledge of the underlying system.

Definition 2 *A discrete-time Parametric Markov chain is defined as a tuple* $\mathbf{M}_\Theta = (S, \mathbb{T}_\theta, \iota_{init}, \mathrm{AP}, L, \Theta)$ *where* $S, \iota_{init}, \mathrm{AP}, L$ *are as in Definition 1. The entries in* $\mathbb{T}_\theta$ *are specified in terms of parameters, collected in a parameter vector* $\theta \in \Theta$*, where* Θ *is the set of all possible evaluations of* θ*. Each evaluation gives rise to an induced Markov chain* $\mathbf{M}(\theta)$*.*

Note that we require a certain type of well-posedness of the parameterisation, in that we demand that $\forall s \in S, \forall \theta \in \Theta : \sum_{s' \in S} \mathbb{T}_\theta(s, s') = 1$. More precisely, for any given $\theta \in \Theta$ the parameterised transition function $\mathbb{T}_\theta$ reduces to a stochastic

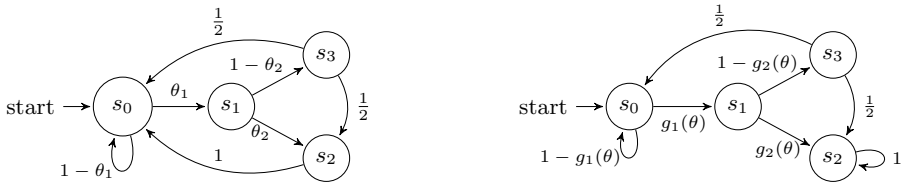


Fig. 1: Two parameterised Markov chains. The nodes of the graph represent states. The labels over the edges provide the probability of taking a transition. The left graph gives parameterised MC with a *basic* parameterisation, where the parameters θ_1, θ_2 are encompassed in the vector $\theta = (\theta_1, \theta_2) \in \Theta = [0,1]^2$. The right graph has a *linear* parameterisation, characterised by affine functions $g_{1,2} : \theta \mapsto [0,1]$.

matrix, and the parametric Markov chain reduces to a Markov chain denoted as $\mathbf{M}(\theta)$ with a probability measure $\mathbb{P}_{\mathbf{M}(\theta)}$.

In the remainder we considered two types of parameterised Markov chain. We use the first, simpler type, as a base case to build the method for the more complex linearly parameterised Markov chains.

1. *basic parameterised Markov chains* with independently parameterised transition probabilities. Consider $\mathbf{M}_\Theta = (S, \mathbb{T}_\theta, \iota_{init}, \mathrm{AP}, L, \Theta)$ with $\Theta \subseteq [0,1]^n$ and parameter vector $\theta := (\theta_1, \ldots, \theta_n) \in \Theta$ build up based on individual parameters $\theta_i \in [0,1]$. Then the parameterised MC is considered *basic* if transition probabilities between states are either known and considered constant with a value in $[0,1]$, or have a single parameter θ_i (or $1-\theta_i$) associated to them and $\forall s \in S, \forall \theta \in \Theta : \sum_{s' \in S} \mathbb{T}_\theta(s, s') = 1$ (cf. Fig. 1, left).
2. *linearly parameterised Markov chains*, where unknown transition probabilities can be linearly related. Given $\Theta \subseteq [0,1]^n$ and parameter vector $\theta := (\theta_1, \ldots, \theta_n) \in \Theta$ with $\theta_i \in [0,1]$, the parameterised MC is considered *linearly parameterised* if there exists a set of affine functions $g_l(\theta) := k_0 + k_1\theta_1 + ... + k_n\theta_n$ with $k_i \in [0,1]$ and $\sum k_i \leq 1$, denoted $g_l(\theta)_{l \in \mathcal{L}}$. All outgoing transition probabilities of states (or, graphically labels of outgoing edges of a node, cf. Fig.1) have probability $g_l(\theta)$ or $1 - g_l(\theta)$ and $\forall s \in S, \forall \theta \in \Theta : \sum_{s' \in S} \mathbb{T}_\theta(s, s') = 1$.

The basic case leads to simple procedures, and in Section 5 we develop the linear structure for Bayesian verification. Parameterisations beyond these two categories, such as *nonlinear* ones, are out of the scope of this paper.

2.2 Properties – Probabilistic Computation Tree Logic

We consider system requirements specified in probabilistic logics. As we leverage PRISM's parametric model checking tool [13] for synthesis, we may consider the same set of properties that the synthesis tool supports. In the case of PRISM, this is non-nested Probabilistic Computational Tree Logic (PCTL) [16] formulae, and so in our work we focus on these properties. For instance, $\mathbb{P}_{\geq 0.5}(\text{stay } \mathcal{U} \text{ get})$

expresses the property that "the probability of remaining in a state labelled with atomic proposition 'stay' until we reach a state labelled as 'get', is bigger or equal to 0.5". PRISM also supports nested PCTL with some restrictions, and a planned extension to this work is to use PROPHESY [10] for parameter synthesis, which supports conditional probabilities and unbounded-time properties. We next define PCTL in nexus to finite discrete-time Markov chains:

Definition 3 *Let a discrete-time Markov chain be given. Let ϕ be a formula that is interpreted over states $s \in S$, and φ be a formula that is interpreted on paths of the DTMC. Also, let $\bowtie \in \{<, \leq, \geq, >\}$, $n \in \mathbb{N}$, $p \in [0,1]$, $c \in AP$. The syntax of PCTL is given by:*

$$\phi := \text{True} \mid c \mid \phi \wedge \phi \mid \neg\phi \mid \mathbb{P}_{\bowtie p}(\varphi), \qquad \varphi := \bigcirc \phi \mid \phi \,\mathcal{U}\, \phi.$$

Definition 4 *Let $\mathbf{M}(\theta)$ be an induced Markov chain of the parametric Markov chain $\mathbf{M}_\Theta$ indexed by parameter $\theta \in \Theta$, and let ϕ be a formula in PCTL. The satisfaction function $f_\phi : \Theta \to \{0,1\}$ where $f_\phi(\theta) = 1$ if $\mathbf{M}(\theta) \models \phi$, and 0 otherwise.*

Let us assume that the satisfaction function f_ϕ is measurable and entails a decidable verification problem for all $\theta \in \Theta$. The computation of the satisfaction function, or equivalently the exploration of a parameter set over a formal property, has been studied for various reactive models in [3, 11, 18].

2.3 Bayesian inference

Our method uses Bayesian inference to learn the probability distribution of parameters in our model class as more evidence or data becomes available. Bayesian inference derives the posterior probability distribution from a prior probability and a likelihood function derived from a statistical model for the observed data. Bayes' law states that, given observed data D the posterior probability of a hypothesis $p(H \mid D)$, is proportional to the likelihood $p(D \mid H)$, multiplied by the prior $p(H)$, as

$$p(H \mid D) = \frac{p(D \mid H)p(H)}{p(D)}. \tag{1}$$

In our work data D comprises batches of traces of specific length generated by Markov chains instantiated over Θ. The term at the denominator in (1) entails an integral over the parameter set Θ, which in general requires numerical approximation. It is of interest to seek a *conjugate* prior $p(H)$ resulting in a closed-form expression for the posterior $p(H \mid D)$: in this work we make use of the Dirichlet distribution, which is conjugate to the multinomial one [30]. However, when insufficient initial knowledge is available, we choose a non-informative prior, which has minimal influence on the posterior, such as a uniform prior.

3 Problem statement and overview of the approach

Consider a partly unknown dynamical system $\mathbf{S}$, and suppose that we can gather a limited amount of sample trajectories from this system as data. Assume that the knowledge about the system is encompassed within a parametric model class, describing the behaviour of $\mathbf{S}$ up to the unknown parameterisation of some of its transitions. We plan to investigate the following goal: *can we efficiently use the gathered data and the model knowledge of* $\mathbf{S}$ *to formally verify given PCTL properties over* $\mathbf{S}$*, quantifying a confidence in our assertions?*

The three phases of our work are as follows. In the first phase, Sec. 4, we use parameter synthesis to determine a set of feasible parameters for which the system satisfies the given property. The second phase, Sec. 5, executed in parallel, uses Bayesian Inference to infer a distribution over the likely value of the parameters given sample data from the system. In the final phase, Sec. 6, the combination of parametric inference and parameter synthesis to quantify the confidence that the system verifies a PCTL property of interest.

Bayesian probability calculus [23] leads to express the confidence in a property as a measure of the uncertainty distribution over the synthesised parameter sets. Uncertainty distributions are handled as probability distributions of random variables. Given a specification ϕ and a data set D, the confidence that $\mathbf{S} \models \phi$ can be quantified via inference as $\mathbb{P}(\mathbf{S} \models \phi \mid D) = \int_{\Theta} f_{\phi}(\theta) p(\theta \mid D)\, d\theta$, where $\mathbb{P}(\cdot)$ is a probability measure obtained integrating the distribution $p(\cdot)$ of the uncertainty parameter over $\mathbf{M}_{\Theta}$, expressed as the *a-posteriori* $p(\theta \mid D)$ given the data set D and the uncertainty distribution $p(\theta)$ over the parameter set Θ.

The computation in the third phase is a key challenge in the case of Markov chains that are parameterised non trivially due to the required complex manipulation of Dirichlet posterior distributions. This motivates the introduction of a Markov chain expansion algorithm in Sec. 5.2, which enables us to analytically obtain samples of complex posterior distributions for the final step of the approach.

4 Parameter synthesis

The first phase of our method uses parameter synthesis and, given a property and a parameterised Markov chain, synthesises the feasible set of parameters corresponding to models satisfying the given PCTL property. This corresponds to the set of parameters for which the binary satisfaction function, $f_{\phi}(\theta) = \mathbb{P}(\mathbf{M}(\theta) \models \phi)$, is equal to 1. We denote this set Θ_{ϕ}, namely

$$\Theta_{\phi} = \{\theta \in \Theta : \mathbf{M}(\theta) \models \phi\}.$$

We leverage PRISM's parametric model checking functionality based on [14, 15] to perform this synthesis. [14, 15] expresses quantitative specifications as rational functions that are later manipulated. PRISM's parameteric model checking

approach can be applied to unbounded until, steady-state probabilities, reachability reward and steady-state reward properties for parametric DTMCs. The result is a mapping from hyper-rectangles (subsets of parameter valuations) to functions over the parameters.

Alternatives to these techniques have not shown to be scalable or sufficiently general. [6] explores the parameter space with the objective of model verification. [20] employs an analytical approach to parameter synthesis for probabilistic transition systems and is bound to at most two parameters. [9] employs a language-theoretical approach based on regular expressions, which however does not scale as the number of transitions of the Markov model increases. [29] synthesise single-parameter Markov models via accurate interval propagation.

5 Bayesian inference in parameterised Markov chains

In this section we consider the application of Bayesian inference to parameterised Markov chains, in order to infer unknown parameter probabilities based on observed data. We will first present the technique for basic parameterised Markov chains, and the method is then detailed for linearly related parameterisations in Sec. 5.2, where we show that data obtained from a linearly parameterised Markov chain can equally be represented by data complemented with a set of hidden (or unobserved) data of a basic Markov chain.We use $\mathbb{P}(\cdot)$ to denote probability measures, and $p(\cdot)$ to denote probability distributions.

5.1 Basic parameterised Markov chains

Let us consider a basic parameterised Markov chain $\mathbf{M}_{\Theta} = (S, \mathbb{T}_{\theta}, \iota_{init}, \mathrm{AP}, L, \Theta)$ (cf. Definition 2). In this basic parameterised Markov chain, every individual parameter θ_i of vector $\theta = (\theta_1, \theta_2, \ldots, \theta_n) \in \Theta$ is exclusively used to assign the outgoing transition probabilities of a single state. We can decompose our parameter vector θ into sub-vectors θ_{s_i}, giving the parameters for the outgoing transitions of the corresponding state s_i. The concatenation of these sub-vectors gives us again θ.

Consider the parameter vector composed of one parameter, $\theta_{s_k} = \theta_j$, and the corresponding state $s_k \in S$, with outgoing transitions θ_j and $1-\theta_j$ to states (say) s_1 and s_2, respectively. We denote by $p(\theta_j)$ the prior over θ_j, which fully defines the transition probabilities $\mathbb{T}_{\theta}(s_k, \cdot)$ at state s_k. Denote a data set D giving transition counts for sample trajectories generated from the real system $\mathbf{S}$. For any pair $(s_k, s_l) \in S \times S$ the number of transitions $s_k \to s_l$ in D is denoted as $D_{s_k}^{s_l}$. The posterior density $p(\theta_j \mid D)$ over θ_j based on D is

$$p(\theta_j \mid D) = \frac{\mathbb{P}(D \mid \theta_j) p(\theta_j)}{\mathbb{P}(D)} = \frac{p(\theta_j) \prod_{s' \in S} \mathbb{T}_{\theta}(s_k, s')^{D_{s_k}^{s'}}}{\mathbb{P}(D_{s_k})} \tag{2}$$

and depends only on $D_{s_k} = \{D_{s_k}^{s'}\}_{s' \in S}$, i.e., the counts of transitions leaving state s_k. Note that the likelihood function $\prod_{s' \in S} (\mathbb{T}_{\theta}(s_k, s'))^{D_{s_k}^{s'}}$ takes the form

of a multinomial distribution.[3] In the case of two outgoing transitions, this is a 2-dimensional multinomial distribution, or a binomial distribution. A closed-form expression for the posterior is obtained by taking a conjugate prior, which, for the class of multinomial distributions, is a Dirichlet distribution with the same number of dimensions as the likelihood function. For the pair $(\theta_j, 1-\theta_j)$ the Dirichlet distribution with hyperparameters $\alpha = (\alpha_1, \alpha_2)$ has a probability density function given by

$$\mathrm{Dir}(\theta_j \mid \alpha) = \tfrac{1}{B(\alpha)} \theta_j^{\alpha_1 - 1} (1-\theta_j)^{\alpha_2 - 1}$$

on the open simplex defined by $0 < \theta_j < 1$. The normalising constant, $B(\alpha)$, is a multinomial beta function, and can be written in terms of gamma functions as $B(\alpha) = \Gamma(\alpha_1)\Gamma(\alpha_2)/\Gamma(\alpha_1 + \alpha_2)$ [8]. Hence, for a prior $p(\theta_j) = \mathrm{Dir}(\theta_j \mid \alpha)$ we obtain the posterior distribution for $\theta_j \sim p(\theta_j \mid D) = \mathrm{Dir}(\theta_j \mid D_{s_k} + \alpha)$, namely

$$p(\theta_j \mid D) \propto p(\theta_j) \textstyle\prod_{s' \in S} \mathbb{T}_\theta(s_k, s')^{D^{s'}_{s_k}} \propto \theta_j^{\alpha_1 - 1}(1-\theta_j)^{\alpha_2 - 1} \theta_j^{D^{s_1}_{s_k}} (1-\theta_j)^{D^{s_2}_{s_k}} \quad (3)$$

where the normalisation constant of the obtained Dirichlet distribution is $B(\alpha + D_{s_k}) = \Gamma(\alpha_1 + D^{s_1}_{s_k})\Gamma(\alpha_2 + D^{s_2}_{s_k})/\Gamma(\alpha_1 + D^{s_1}_{s_k} + \alpha_2 + D^{s_2}_{s_k})$. In other words, as data is gathered, we analytically update the posterior probability distribution $p(\theta_j \mid D)$ by updating the parameters of a Dirichlet distribution.

This result can be extended to the case of a state s_l with $m > 2$ outgoing transitions. We parameterise the outgoing transitions with the sub-vector $\theta_{s_l} = (\theta_1, ..., \theta_{m-1})$ and $1 - \theta_1 - ... - \theta_{m-1}$, and obtain the posterior for the sub-vector, $p(\theta_{s_l} \mid D)$. The likelihood function takes the form of an m-dimensional multinomial distribution, and we express the prior as an m-dimensional Dirichlet.[4] This yields a posterior distribution as an m-dimensional Dirichlet distribution, $p(\theta_{s_l}|D) = \mathrm{Dir}(\theta_{s_l} \mid D_{s_l} + \alpha)$.

The posterior distribution for the entire parameter vector $p(\theta \mid D)$ is equal to the product of the posterior distributions for the sub-vectors of θ. This holds due to the stated independence of the parameters in a basic parameterised Markov chain, which results in independent priors and independent likelihood functions. Hence $p(\theta \mid D) = \prod_{s_i} \mathrm{Dir}(\theta_{s_i} \mid D_{s_i} + \alpha)$.

Transition grouping. For simplicity, given a state with multiple outgoing transitions we may obtain the distribution for each parameter using marginal distributions. Consider state s_l with $m > 2$ outgoing transitions, parameterised with the sub-vector $\theta_{s_l} = (\theta_1, ..., \theta_{m-1})$ and $1 - \theta_1 - ... - \theta_{m-1}$ We have shown earlier that, if the parameters are independent, the joint posterior distribution

[3] A multinomial is defined by its density function $f(\cdot \mid p, N) \propto \prod_{i=1}^{k} p_i^{n_i}$, for $n_i \in \{0, 1, ..., N\}$ and such that $\sum_{i=1}^{k} n_i = N$, where $N \in \mathbb{N}$ is a parameter and p is a discrete distribution over k outcomes.

[4] An m-dimensional Dirichlet distribution is given by $f(x_1, ...x_m; \alpha_1, ..., \alpha_m) = \frac{1}{B(\alpha)} \prod_{i=1}^{m} x_i^{\alpha_i - 1}$ on the open $(m-1)$ dimensional simplex defined by $x_1, ..., x_{m-1} > 0$, $x_1 + ... + x_{m-1} < 1$ and $x_m = 1 - x_1 - ... - x_{m-1}$.

over the transition probabilities for this state is an m-dimensional Dirichlet: $p(\theta_{s_l}|D) = \mathrm{Dir}(\theta_{s_l} \mid D_{s_l} + \alpha)$. The marginal distribution of θ_i is a 2-dimensional Dirichlet, or a beta distribution, $\theta_i \sim \mathrm{Dir}(\alpha_i, (\sum_{i=1}^{m} \alpha_i) - 1)$. We can hence obtain a posterior distribution for each parameter, and we note that we are effectively grouping the training data together for all transitions except the one we obtain the posterior distribution for.

5.2 Linearly parameterised Markov chains

In this section we build on the Bayesian inference for basic parameterisations and tackle linearly parameterised Markov chains. As defined before, in a linear parameterised Markov chain, the transition probabilities will be expressed in the form $g(\theta) = k_0 + k_1\theta_1 + ... + k_n\theta_n$. For a given data set D and a linearly parameterised Markov chain we want to use Bayesian inference to get the posterior distribution $p(\theta|D)$ over the parameter set Θ. In order to work with linear parameters we introduce two types of transformations of the Markov chain. In the first we show that by introducing additional, non-observed states, into the Markov chain and the data, the linear parameterised Markov chain can be transformed to a basic Markov chain with unobserved states (and hidden data). Next to that, we consider a compression of the data. When two states of the DTMC have "similar" transitions, what can be learned is equivalent. These states are referred to as being *parameter similar* and will be introduced more precisely in the following. After these transformations we can apply the Bayes rule over the expanded Markov chain and hidden data.

Parameter similar states. If we have the same parameter appearing multiple times in our Markov chain, we must combine the data obtained from all these transitions to obtain a sole posterior distribution for the parameter to be inputted into our confidence computation. This technique, referred to as "parameter tying", is used in [26]. We can perform this step analytically for Dirichlet distributions over *parameter similar states*, by which we denote states with outgoing transitions having identical parameterisations.

Manipulating posterior Dirichlet distributions is mathematically complex because of the dependence between the variables. However, if states are parameter similar, we can use the result in (3). Consider two parameter similar states, s_1 and s_2, with outgoing transition probabilities θ_j and $1 - \theta_j$, and observed data over the transitions. We combine the data to give one posterior Dirichlet distribution for the parameter, $p(\theta_j) = \mathrm{Dir}(D_{s_1} + D_{s_2} + \alpha_{s_1})$.

Parameterised Markov chain state expansions. Consider a parameterised DTMC $\mathbf{M}_\Theta = (S, \mathbb{T}, \iota_{init}, \mathrm{AP}, L, \Theta)$. We wish to define a new parameterised DTMC $\mathbf{M}^*_\Theta$ that produces the same output for our method, but which has a simpler parameterisation. Our method hinges on obtaining a distribution for θ based on collected training data D, and so if $\mathbf{M}^*_\Theta$ is equivalent to $\mathbf{M}_\Theta$, the probabilities of reaching a set of states in $\mathbf{M}_\Theta$ must be the same as reaching

the equivalent states in $\mathbf{M}^*_\Theta$, but we may disregard the length of associated paths. Before introducing the definition of state expansion, we first need to define hidden data. Suppose the two Markov chains have states S and S^*, such that $S \subset S^*$: all states of S^* not in S are defined as *hidden*, i.e., $S^* \setminus S$ are hidden states. Ω denotes the set of finite paths ω in $\mathbf{M}_\Theta$, and Ω^* denotes the set of finite paths ω^* in $\mathbf{M}^*_\Theta$. Then any observed state sequence consists only of states in S, whereas the states in $S^* \setminus S$ remain hidden from the observations. The data set D over the states S consists of transition counts $D^{s_l}_{s_k}$ for pairs $s_k, s_l \in S$. Observe that for the set of states S^* the data is incomplete, namely it does not represent the actual state transitions but only the observed ones. For an observed transition count $D^{s_l}_{s_k}$, we introduce the extended set $D^{s_l\,*}_{s_k}$ as the collection of counts over all hidden paths from s_k to s_l. For hidden states s^*_1 and s^*_2 (labeled for convenience with $*$), hidden paths over $s_k, s_l \in S$ can be of the form $\{s_k, s_l\}, \{s_k, s^*_1, s_l\}, \{s_k, s^*_1, s^*_2, s_l\}, \{s_k, s^*_2, s_l\} \in \Omega^*$, with the associated extended data count $D^{s_l\,*}_{s_k} := \{D^{s^*_1}_{s_k}, D^{s^*_2}_{s_k}, D^{s^*_2}_{s^*_1}, \ldots\}$. The set of possible extended transition counts is denoted as $\mathcal{D}^{s_l\,*}_{s_k}$ for the pair (s_k, s_l), and $\mathcal{D}^*$ for all transitions – note that they are set-valued mappings of $D^{s_l}_{s_k}$ and D, respectively.

Definition 5 *Consider parameterised Markov chains* $\mathbf{M}_\Theta = (S, \mathbb{T}, \iota_{init}, \mathrm{AP}, L, \Theta)$ *and* $\mathbf{M}^*_\Theta = (S^*, \mathbb{T}^*, \iota^*_{init}, \mathrm{AP}, L^*, \Theta)$*, both over set* Θ*. We say that* $\mathbf{M}^*_\Theta$ *is* an expansion *of* $\mathbf{M}_\Theta$ *if, for all* D *and for all* $\theta \in \Theta$*,*

$$\mathbb{P}_{\mathbf{M}(\theta)}(D) = \mathbb{P}_{\mathbf{M}^*(\theta)}(\mathcal{D}^*),$$

and if $\iota_{init} = \iota^*_{init}$*. The extended labelling map* L^* *is a trivial extension of* L*, assigning labels* $L(s)$ *for* $s \in S$ *and an empty label to* $S^* \setminus S$*.*

Theorem 1. *The expansion relation is transitive; if* $\mathbf{M}_{\Theta,1}, \mathbf{M}_{\Theta,2}, \mathbf{M}_{\Theta,3}$ *are all parameterised with* Θ*,* $\mathbf{M}_{\Theta,3}$ *is an expansion of* $\mathbf{M}_{\Theta,2}$ *and* $\mathbf{M}_{\Theta,2}$ *is an expansion of* $\mathbf{M}_{\Theta,1}$*, then* $\mathbf{M}_{\Theta,3}$ *is an expansion of* $\mathbf{M}_{\Theta,1}$*.*

Case I: Transition splitting. We split a transition probability parameterised with $k_0 + \sum_i k_i\theta_i$ into transitions to hidden states with probabilities $k_i\theta_i$, and refer to this operation as *transition splitting*. As a basic example, consider Fig. 2 where state s_0 in M has two outgoing transition probabilities expressed as functions of the parameter vector, $g(\theta)$ and $1 - g(\theta)$, where $g(\theta) = k_0 + k_1\alpha + k_2\beta$. We expand $\mathbf{M}_\Theta$ into $\mathbf{M}^*_\Theta$ by splitting state s_1 into a set of states, and splitting the transition from $s_0 \to s_1$ into the monomials concerning each parameter in θ, as shown in Fig. 2. $\mathbf{M}^*_\Theta$ is an expansion of $\mathbf{M}_\Theta$ as per Def. 5.

Lemma 1. *Transition splitting of* $\mathbf{M}_\Theta$ *(Case I) generates an expansion of* $\mathbf{M}_\Theta$*.*

Case II: State splitting. We present a second case, state splitting, for a parameter θ_i multiplied by a constant, $k_i\theta_i$. Consider the simple DTMC in Fig.3a, and the state s_0 in $\mathbf{M}_\Theta$ with two outgoing transition probabilities expressed as a constant multiplied by one parameter, $k_1\theta_1$ and $1 - k_1\theta_1$, where $0 \leq k_1 \leq 1$. We expand $\mathbf{M}_\Theta$ to give $\mathbf{M}^*_\Theta$ by splitting state s_0 into two states, and compute the

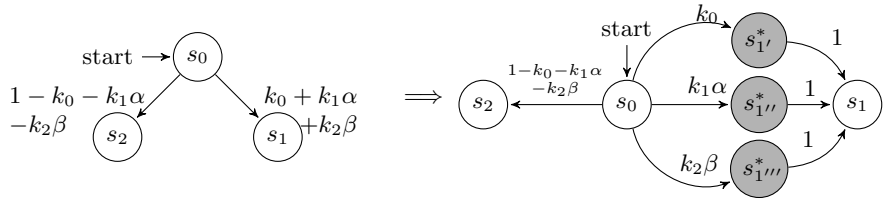
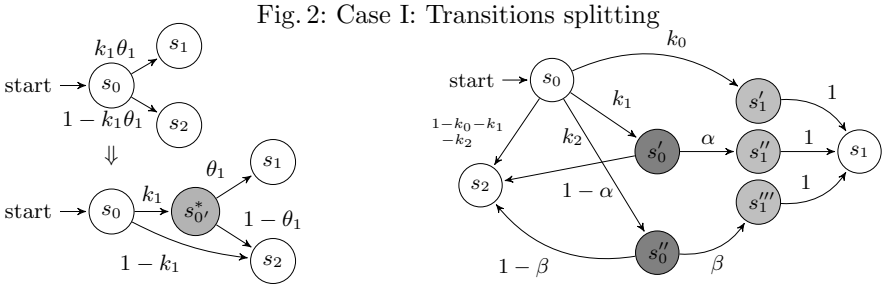


Fig. 2: Case I: Transitions splitting



(a) Simple example of state splitting (b) State splitting of Fig.2 (cf. Case I).

Fig. 3: Case II: state splitting (two examples)

transition probabilities that the imposed expansion demands. As an additional example, notice that the transitions studied in Case I are all of the form $k_i\theta_i$. Applying the state splitting to this expanded DTMC we obtain Fig. 3b. The subsequent application of both state splitting cases (cf. Fig. 3b) induces again an expanded parameterised Markov chain as per Def. 5.

Lemma 2. *State splitting of $\mathbf{M}_\Theta$ (Case II) generates an expansion of $\mathbf{M}_\Theta$.*

We are led to the following result.

Theorem 2. *Any linearly parameterised Markov chain can be expanded into a basic parameterised Markov chain by application of Lemma 1 and 2.*

Bayesian inference with missing data We now consider Bayesian inference on the newly expanded Markov chain $\mathbf{M}^*_\Theta$. The data set D, which is sampled from our system, corresponds to a state trajectory or set of trajectories over the model $\mathbf{M}_\Theta$. This set further comprises only part of the corresponding trajectories in the expanded model $\mathbf{M}^*_\Theta$. For a given trajectory in D, we refer to D^* as the completed trajectory, and to $\mathcal{D}^*$ as the set of all possible completions D^*. Note that the expanded parametric Markov chain has a *basic parameterisation*, hence for a given completed data set D^* the Bayes rule as elaborated in (1) can be applied to obtain $p(\theta|D^*)$. For $\mathbf{M}^*_\Theta$ Bayes rule can be applied over the hidden data as follows:

$$p\left(\theta|D\right) = \frac{\sum_{D^*\in\mathcal{D}^*} p\left(\theta, D^*, D\right)}{\mathbb{P}(D)} = \frac{\sum_{D^*\in\mathcal{D}^*} p\left(\theta \mid D^*, D\right)\mathbb{P}(D^* \mid D)\mathbb{P}(D)}{\mathbb{P}(D)}$$
$$= \sum_{D^*\in\mathcal{D}^*} p\left(\theta|D^*\right)\mathbb{P}(D^*|D).$$

Completed data sets have a multinomial distribution dependent on the parameterisation, hence the distribution of D^* is given as $\mathbb{P}(D^*) = \int_\Theta \mathbb{P}(D^*|\theta) p\,(\theta)\, d\theta$. For a given D the conditional distribution $\mathbb{P}(D^*|D)$ is $\mathbb{P}(D^*|D) = \mathbb{P}(D^*)/\mathbb{P}(D)$, with $D^* \in \mathcal{D}^*$ and $\mathbb{P}(D) = \sum_{\mathcal{D}^*} \int_\Theta \mathbb{P}(D^*|\theta) p\,(\theta)\, d\theta$.

Remark 1. Realisations of the posterior can be obtained without computing the entire integral as follows. A set of realisations θ_i for $i \in \{1, \ldots, \mathcal{N}\}$ with probability density function $p\,(\theta|D)$ can be obtained by generating samples D_i^* with distribution $\mathbb{P}(D^*|D)$ and subsequently generating samples θ_i with distribution $p\,(\theta|D_i^*)$ for all $i \in \{1, \ldots, \mathcal{N}\}$. These samples can then directly be used to perform the confidence calculation as in Sec. 6. □

Algorithm 1 presents the state expansion procedure, and Algorithm 2 in the next section summarises how to obtain a realisation of the posterior $p(\theta \mid D^*)$, and to integrate it with the confidence computation.

Algorithm 1 Markov chain expansion ($\mathbf{M}_\Theta$)

$\mathbf{M}^*_\Theta \leftarrow \mathbf{M}_\Theta$
for all $s_i \in S^*$ **do** ▷ Case I: transition splitting
  for all $\mathbb{T}^*_\theta(s_i, s_j) = k_0 + \sum_{l \in \mathcal{L}} k_l \theta_l$ **do**
    $S^* \leftarrow \{s^*_{ij,l}\}_{l \in L} \cup s_{ij,0}$
    $\mathbb{T}^*_\theta(s_i, s_j) := 0$
    $\mathbb{T}^*_\theta(s_i, s^*_{ij,0}) := k_0$ and $\mathbb{T}^*_\theta(s^*_{ij,0}, s_j) := 1$
    for all $l \in L$ **do**
      $\mathbb{T}^*_\theta(s_i, s^*_{ij,l}) := k_l \theta_l$ and $\mathbb{T}^*_\theta(s^*_{ij,l}, s_j) := 1$
for all $s_i \in S^*$ **do** ▷ Case II: state splitting
  if $\exists s_k \in S^* : \mathbb{T}^*_\theta(s_i, s_k) = 1 - k_0 - \sum_{l \in \mathcal{L}} k_l \theta_l$ **then**
    $\mathbb{T}^*_\theta(s_i, s_k) := 1 - k_0 - \sum_{l \in \mathcal{L}} k_l$
    for all $\mathbb{T}^*_\theta(s_i, s_m) = k_l \theta_l$ **do**
      $S^* \leftarrow s^*_{m'}$
      $\mathbb{T}^*_\theta(s_i, s_m) := 0$, $\mathbb{T}^*_\theta(s_i, s^*_{m'}) := k_l$ and $\mathbb{T}^*_\theta(s^*_{m'}, s_k) := 1 - \theta_l$
      $\mathbb{T}^*_\theta(s^*_{m'}, s_m) := \theta_l$
return $\mathbf{M}^*_\Theta$ ▷ return expanded DTMC

6 Bayesian verification: computation of confidence

In this section we detail the final phase of our method: a quick procedure computes a confidence estimate for the satisfaction of a PCTL specification formula ϕ by a system $\mathbf{S}$ of interest, namely $\mathbf{S} \models \phi$. Our method takes as input a posterior distribution over Θ, obtained using Bayesian inference in Sec. 5.2, and the feasible set for the parameters, obtained by parameter synthesis in Sec. 4.

Definition 6. *Given a PCTL specification ϕ, a complete trace (sample trajectory) D of the system $\boldsymbol{S}$ up to time t, and a transition function $\mathbb{T}$, the confidence that $\boldsymbol{S} \models \phi$ can be quantified by Bayesian Inference as*

$$\mathbb{P}(\boldsymbol{S} \models \phi \mid D) = \int_\Theta f_\phi(\theta) p(\theta \mid D) d\theta. \tag{4}$$

As we only consider the satisfaction of a property $\mathbf{S} \models \phi$ as a binary-valued mapping from the space of parameters, the satisfaction function in (4), $f_\phi : \Theta \to \{0, 1\}$, (4) can be reformulated as:

$$\mathbb{P}(\mathbf{S} \models \phi \,|\, D) = \int_{\Theta_\phi} p(\theta \mid D) d\theta, \tag{5}$$

where Θ_ϕ denotes the set of parameters corresponding to models verifying the property ϕ (as generated by PRISM). Further, given the independent posterior distributions for each parameter in θ resulting from Sec. 5.2, the confidence can be computed as $\mathbb{P}(\mathbf{S} \models \phi \,|\, D) = \int_{\Theta_\phi} \prod_{\theta_i \in \theta} p(\theta_i \mid D) d\Theta$. The integral of a Dirichlet distribution can be obtained by iterative or numerical methods: here we use a simple Monte-Carlo approach, which depends on samples of the posterior distribution as clarified in Algorithm 2.

Algorithm 2 Monte-Carlo Integration for linearly parameterised DTMC

$\mathcal{N}$:= number of Monte-Carlo samples
$\{D_i^*\}_{i \in \{1,...,\mathcal{N}\}} \sim p(D^*|D)$ ▷ hidden data samples
for all $i \in \{1, ..., \mathcal{N}\}$ **do**
  Compute $p(\theta|D_i^*)$ ▷ Bayesian inference
  $\theta_i \sim p(\theta|D_i^*)$ ▷ posterior samples
  $j_\# \leftarrow j_\# + \text{Boolean}[\theta_i \in \Theta_\phi]$
$\hat{\mathbb{P}}(\mathbf{S} \models \phi) := \frac{j_\#}{\mathcal{N}}$
return $\hat{\mathbb{P}}(\mathbf{S} \models \phi)$ ▷ estimate of $\mathbb{P}(\mathbf{S} \models \phi)$

7 Experiment results

We show that our approach requires smaller amounts of data than standard statistical model checking (SMC) to verify the system satisfies a given quantitative specification up to a prescribed confidence level. We further claim that our approach is more robust than standard SMC in situations where only data of limited trace length is available.

Experiment setup We focus our experimental discussion on the basic parameterised Markov chain $\mathbf{M}_\Theta$ in Figure 1 and the PCTL property $\phi = \mathbb{P}_{>0.5}[\neg s_3 \,\mathcal{U}\, s_2]$, where we have employed identity labels to decorate states.

We note that the ground truth for $\mathbf{S} = \mathbf{M}(\theta)$, namely Y_{true}, is a step function over the parameter θ, namely

$$Y_{true} = \begin{cases} 0 & \text{if } \theta \leq 0.5 \\ 1 & \text{if } \theta > 0.5, \end{cases} \tag{6}$$

so that the feasible set is $\Theta_\phi = [0.5, 1]$. We choose a uniform prior for both methods: for our approach $p(\theta \mid D) = \text{Dir}(1, 1)$, which, for property ϕ, means $p(\mathbf{M}(\theta) \models \phi) = \text{Dir}(1, 1)$; for SMC we set $p(\mathbf{M}(\theta) \models \phi) = \text{Dir}(1, 1)$. We run

both methods over empirical data obtained from $\mathbf{M}(\theta)$ for varying values of θ, and compare the outcomes with the ground truth.

The core idea of the SMC of interest for this work is to collect sample trajectories from the system, to then determine whether the trajectories satisfy a given property, and to apply statistical techniques (such as hypothesis testing) to decide whether the system satisfies the property or not, with some degree of confidence. Our approach collects data from the system, uses the data to determine a distribution over parameter values in the parameterised model class and applies statistical techniques (in this case, a Bayesian confidence calculation) to decide whether the system satisfies the property or not, with some degree of confidence. We could then additionally apply hypothesis testing to our approach. However, as we do not do this, for a meaningful comparison with our approach we implement the early steps of the SMC procedure from [32], and omitting the hypothesis testing we compute a Bayesian confidence by integrating the posterior distribution given over the [0,1] interval, representing the probability of a trace satisfying the property. The trace generation and trace verification stages of SMC are implemented in the same way in the four statistical model checking methods in PRISM.

We collect training data from our original system in the form of a history of states visited up until time t, denoted again as D. We use $100 \leq |D| \leq 200,000$ state transitions in finite traces, e.g., D contains traces of length 100, e.g., when $|D| = 1000$ we have 10 traces of length 100. We run both methods with the same data.We test robustness to trace length, e.g., when $|D| = 1000$, we may have 100 traces of length 10, or 10 traces of length 100.

We compute the mean squared error (MSE) between the confidence outcome and the ground truth from Equation (6), namely $MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_{true} - Y_i)^2$, where n is the number of experiments run and Y_i is the result $\mathbb{P}(\mathbf{M}_\theta \models \phi)$ for the i-th run.

We disregard the numerical error in the Monte Carlo approximate integration, which is the same for both techniques. We cover the parameter range $0.3 \leq \theta \leq 0.7$, selected at intervals of 0.05.

Results and Discussion The first point to note is that the confidence is low, and the MSE high for the parameter values close to $\theta = 0.5$ for both approaches. This is due to $\theta = 0.5$ being on the edge of the feasible set and is consistent with the information we wish to obtain from the confidence calculation: if the parameter value is near the edge of the feasible set, we need to know its value precisely to be sure it falls in the feasible set. Consider that in order to compute the confidence of satisfaction of the property ϕ, we integrate the posterior distribution over the feasible set $\Theta_\phi = \{\theta > 0.5\}$. The posterior distribution obtained for $\theta = 0.5$ should have a peak centred at 0.5 and half of the area under the peak should fall in the feasible set, leading to $\mathbb{P}(\mathbf{M}(\theta) \models \phi) = 0.5$. The height and width of the distribution $p(\theta \mid D)$ are characterised by the amount of data available, as well as by the consistency of the data, and so we expect the MSE to be higher for parameter values close to the threshold.

The key result is that the mean squared error reduces as $|D|$ increases and the variance decreases in both approaches, but our approach consistently produces a smaller error and variance than SMC for any parameter values excluding $\theta = 0.5$ (where both approaches perform comparably). Our approach requires an order of magnitude less data than SMC and above $|D| = 2000$, the error for our approach is smaller than the error in the Monte Carlo integration, whereas SMC does not reach this precision threshold in our experiments, which we perform up to $|D| = 200000$.

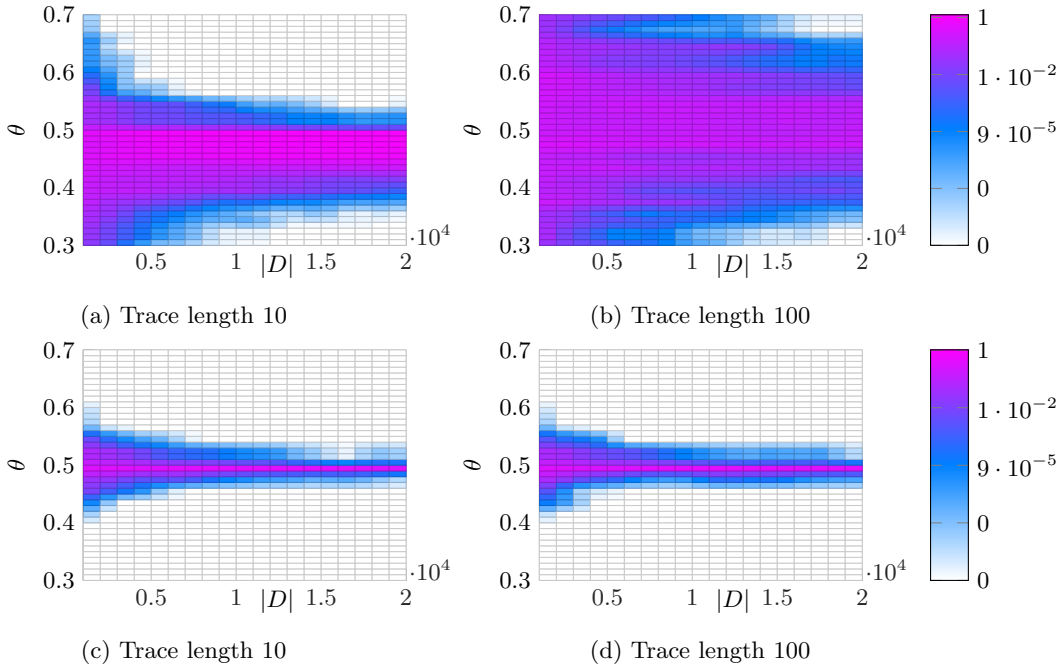
We ascribe both the reduced error and reduced variance to the data efficiency of our approach: SMC receives the training data in the form of short traces, and discerns whether a trace is a counter example or witness for the property. A trace can, however, be neither, in which case it is discarded, even if that trace contains parameterised transitions. Our approach counts each parameterised transition in the training data, and so SMC uses less of the data available than our approach. It is unsurprising that accuracy and variance improve when more data is used.

We investigate robustness in a situation where it is only possible to collect short trajectories from the system, whilst verifying an unbounded property. Figure 4a and Figure 4b show the performance of SMC with $|D|$ made up of trace lengths of 10 and 100 transitions respectively. We show a part of our data set, discarding data above $|D| = 20,000$ where our approach produces no measurable error. The mean squared error in Figure 4b is 50% lower than in Figure 4a over the entire parameter range, but the run with trace lengths of 10 performs better for values of $\theta > 0.55$.

We explain this because, computed using PRISM, the expected length of a witness for our property and Markov Chain ranges between 4.33, for $\theta = 0.3$ and 2.42 for $\theta = 0.7$ (due to the symmetrical structure of our Markov Chain, the lengths of counter-examples are also expected to be the same). Thus a large proportion of the traces of length 10 are discarded, and so SMC has less data to use, explaining the increased error across the parameter range. However, when $\theta > 0.55$, the expected counter-example length is higher, and so the number of traces of length 10 that are useful begins to exceed the total number of traces of length 100 received.

In contrast, the performance of our approach, shown in Figure 4c and Figure 4d,yields approximately the same outcomes for both trace lengths, as we consider each transition in the training data individually and only discard non-parameterised transitions. Admittedly it is not always the case that the performance of our method is independent of the length of the traces: consider for example the case of a large Markov chain where a parameterised transition is only reachable after a large number of steps. In this case the performance of our approach would be comparable to SMC.

We run experiments on linearly parameterised Markov chains of a similar scale and obtain comparable results.



(a) Trace length 10

(b) Trace length 100

(c) Trace length 10

(d) Trace length 100

Fig. 4: Outcomes of SMC are given in (a) and (b), outcomes of our approach are given in (c) and (d). The comparison is done over a data set D composed of traces of 10 and 100 transitions. On the x-axis, $1000 \leq |Dt| \leq 20000$. On the y-axis, $0.3 \leq \theta \leq 0.7$. The darker (purple) colour indicates a higher mean squared error.

8 Conclusions and future work

We have presented a data-based verification approach that addresses incomplete model knowledge. The method offers a framework to integrate Bayesian inference and formal verification, and in comparison to standard statistical model checking promises to be more parsimonious with the required data.

We plan to investigate extensions in the following directions: performing parameter synthesis with alternative available techniques, such as [10], which builds on the work of [14, 15] using graph topological properties and fixed points); working with non-linearly parameterised Markov chains; inspired by [12], integrating external non-determinism in the form of actions, thus leading to parameterised Markov decision processes [13]. Finally, we are interested in the use of Bayesian hypothesis testing, which will further solidify this method as a provable verification technique even when the prior probability distribution is not reliably known.

References

1. C. Baier and J. Katoen. *Principles of model checking*. MIT Press, 2008.
2. E. Bartocci, L. Bortolussi, and G. Sanguinetti. Learning temporal logical properties discriminating ECG models of cardiac arrhytmias. *CoRR*, abs/1312.7523, 2013.
3. G. Batt, C. Belta, and R. Weiss. Model checking genetic regulatory networks with parameter uncertainty. In *HSCC*, volume 4416 of *Lecture Notes in Computer Science*, pages 61–75. Springer, 2007.
4. J. Bernardo and A. Smith. *Bayesian Theory*. Chichester: Wiley, 1994.
5. L. Bortolussi and G. Sanguinetti. Learning and designing stochastic processes from logical constraints. *Logical Methods in Computer Science*, 11(2), 2015.
6. L. Brim, M. Ceska, S. Drazan, and D. Safránek. Exploring parameter space of stochastic biochemical systems using quantitative model checking. In *CAV*, volume 8044 of *Lecture Notes in Computer Science*, pages 107–123. Springer, 2013.
7. Y. Chen and T. D. Nielsen. Active learning of Markov decision processes for system verification. In *ICMLA*, pages 289–294. IEEE, 2012.
8. P. J. Davis. Gamma function and related functions. *Handbook of Mathematical Functions*, 1965.
9. C. Daws. Symbolic and parametric model checking of discrete-time Markov chains. In *ICTAC*, volume 3407 of *Lecture Notes in Computer Science*, pages 280–294. Springer, 2004.
10. C. Dehnert, S. Junges, N. Jansen, F. Corzilius, M. Volk, H. Bruintjes, J. Katoen, and E. Ábrahám. Prophesy: A PRObabilistic ParamEter SYnthesis Tool. In *CAV*, volume 9206 of *Lecture Notes in Computer Science*, pages 214–231. Springer, 2015.
11. G. Frehse, S. K. Jha, and B. H. Krogh. A counterexample-guided approach to parameter synthesis for linear hybrid automata. In *HSCC*, volume 4981 of *Lecture Notes in Computer Science*, pages 187–200. Springer, 2008.
12. S. Haesaert, P. M. J. Van den Hof, and A. Abate. Data-driven property verification of grey-box systems by Bayesian experiment design. In *ACC*, pages 1800–1805. IEEE, 2015.
13. E. M. Hahn, T. Han, and L. Zhang. Synthesis for PCTL in parametric Markov decision processes. In *NASA Formal Methods*, volume 6617 of *Lecture Notes in Computer Science*, pages 146–161. Springer, 2011.
14. E. M. Hahn, H. Hermanns, B. Wachter, and L. Zhang. PARAM: A model checker for parametric Markov models. In *CAV*, volume 6174 of *Lecture Notes in Computer Science*, pages 660–664. Springer, 2010.
15. E. M. Hahn, H. Hermanns, and L. Zhang. Probabilistic reachability for parametric Markov models. In *SPIN*, volume 5578 of *Lecture Notes in Computer Science*, pages 88–106. Springer, 2009.
16. H. Hansson and B. Jonsson. A logic for reasoning about time and reliability. *Formal Asp. Comput.*, 6(5):512–535, 1994.
17. D. Henriques, J. Martins, P. Zuliani, A. Platzer, and E. M. Clarke. Statistical model checking for Markov decision processes. In *QEST*, pages 84–93. IEEE Computer Society, 2012.
18. T. A. Henzinger and H. Wong-Toi. Using HyTech to synthesize control parameters for a steam boiler. In *Formal Methods for Industrial Applications*, volume 1165 of *Lecture Notes in Computer Science*, pages 265–282. Springer, 1995.
19. M. Z. Kwiatkowska, G. Norman, and D. Parker. The PRISM benchmark suite. In *QEST*, pages 203–204. IEEE Computer Society, 2012.

20. R. Lanotte, A. Maggiolo-Schettini, and A. Troina. Parametric probabilistic transition systems for system design and analysis. *Formal Asp. Comput.*, 19(1):93–109, 2007.
21. A. Legay, B. Delahaye, and S. Bensalem. Statistical model checking: An overview. In *RV*, volume 6418 of *Lecture Notes in Computer Science*, pages 122–135. Springer, 2010.
22. A. Legay and S. Sedwards. Lightweight Monte Carlo algorithm for Markov decision processes. *CoRR*, abs/1310.3609, 2013.
23. D. V. Lindley. The philosophy of statistics. *Journal of the Royal Statistical Society: Series D (The Statistician)*, pages 293–337, 2000.
24. H. Mao and M. Jaeger. Learning and model-checking networks of I/O automata. In *ACML*, volume 25 of *JMLR Proceedings*, pages 285–300. JMLR.org, 2012.
25. A. G. Peter Eichelsbacher. Bayesian inference for Markov chains. *Journal of Applied Probability*, 39(1):91–99, 2002.
26. P. Poupart, N. A. Vlassis, J. Hoey, and K. Regan. An analytic solution to discrete Bayesian reinforcement learning. In *ICML*, volume 148 of *ACM International Conference Proceeding Series*, pages 697–704. ACM, 2006.
27. S. Ross, J. Pineau, B. Chaib-draa, and P. Kreitmann. A Bayesian approach for learning and planning in partially observable Markov decision processes. *Journal of Machine Learning Research*, 12:1729–1770, 2011.
28. K. Sen, M. Viswanathan, and G. Agha. Statistical model checking of black-box probabilistic systems. In *CAV*, volume 3114 of *Lecture Notes in Computer Science*, pages 202–215. Springer, 2004.
29. G. Su and D. S. Rosenblum. Nested reachability approximation for discrete-time Markov chains with univariate parameters. In *ATVA*, volume 8837 of *Lecture Notes in Computer Science*, pages 364–379. Springer, 2014.
30. G. Walter. A technical note on the dirichlet-multinomial model, 2012.
31. H. L. S. Younes and R. G. Simmons. Probabilistic verification of discrete event systems using acceptance sampling. In *CAV*, volume 2404 of *Lecture Notes in Computer Science*, pages 223–235. Springer, 2002.
32. P. Zuliani, A. Platzer, and E. M. Clarke. Bayesian statistical model checking with application to Stateflow/Simulink verification. volume 43, pages 338–367, 2013.

A Proofs

A.1 Proof of Theorem 1

Proof. This follows by showing that

$$\mathbb{P}_{\mathbf{M}_1(\theta)}(D) = \mathbb{P}_{\mathbf{M}_2^*(\theta)}(\mathcal{D}^*)$$

is transitive, since transitions generated at a certain state are only dependent on the parameterisation. □

A.2 Proof of Lemma 1

Proof. To show this, consider the paths in $\Omega \in \mathbf{M}$ and $\Omega^* \in \mathbf{M}^*$ that pass through the same states in S in the same order. The probability of the path

$\{s_0 s_2\}$ in $\mathbf{M}$ is trivially the same as the probability of the path $\{s_0 s_2\}$ in $\mathbf{M}^*$. Thus for all $D_{s_0}^{s_2}$, $\mathcal{D}_{s_0}^{s_2} = D_{s_0}^{s_2}$. The probability of the path $\{s_0 s_1\}$ is equal to the sum of the probabilities of the paths $\{s_0 s_{1'}^*, s_1\}, \{s_0 s_{1''}^* s_1\}, \{s_0 s_{1'''}^* s_1\}$. Hence also the probability requirement for $\mathcal{D}_{s_0}^{s_1} = D_{s_0}^{s_2}$ holds and $\mathbf{M}_{\Theta}^*$ is an expansion for $\mathbf{M}_{\Theta}$. Moreover this can be shown to hold for any parameterised transition probability $k_0 + \sum_i k_i \theta_i$. □

A.3 Proof of Lemma 2

Proof. To show this, consider the paths in $\Omega \in \mathbf{M}$ and $\Omega^* \in \mathbf{M}^*$ that pass through the same states in S in the same order. The probability of the path $\{s_0 s_1\}$ in $\mathbf{M}$ is trivially the same as the probability of the path $\{s_0 s_1\}$ in $\mathbf{M}^*$. Thus for all $D_{s_0}^{s_1} : \mathcal{D}_{s_0}^{s_1} = D_{s_0}^{s_1}$. The probability of the path $\{s_0 s_2\}$ is equal to the sum of the probabilities of the paths $\{s_0 s_{0'}^*, s_2\}, \{s_0 s_2\}$. Hence also the probability requirement for $\mathcal{D}_{s_0}^{s_2} = D_{s_0}^{s_2}$ holds and $\mathbf{M}_{\Theta}^*$ is an expansion for $\mathbf{M}_{\Theta}$. Moreover this can be shown to hold for any parameterised transition probability $k_0 + \sum_i k_i \theta_i$. We observe that the expansion holds when $\mathbf{M}_{\Theta}$ is part of a bigger Markov chain, $\mathbf{M}_{\Theta} \subset \mathbf{M}_{full}$, as paths in $\mathbf{M}_{\Theta}$ affected by the expansion will be paths that contain the path fragment $\{s_0 s_1\}$ or $\{s_0 s_2\}$. The path fragments are equal to the set of paths $\Omega \in \mathbf{M}$, and we have shown that these are equivalent to the set of paths $\Omega^* \in \mathbf{M}^*$. Thus the path probabilities in $\mathbf{M}_{\Theta}^*$ are equal to the path probabilities in $\mathbf{M}_{\Theta}$ and the equivalence holds. □

A.4 Proof of Theorem 2

Proof. Consider a linearly parameterised Markov chain $\mathbf{M}_{\Theta}$, with n states. Each state has m outgoing transitions, parameterised with linear functions of sub-vectors $\theta_{s_i} = \theta_1, ..., \theta_k$. We apply transition splitting, Case I, to every transition in the Markov chain. By Lemma 1, we generate an expansion of $\mathbf{M}_{\Theta}$, with n states, nk hidden states and transition probabilities expressed as $k_i \theta_i$ or constants. We apply state splitting, Case II, to every state in the new M_{Θ} with outgoing transitions probabilities expressed as $k_i \theta_i$. By Lemma 1, this generates an expansion of $\mathbf{M}_{\Theta}$, with transition probabilities expressed as constants, or parameters; a basic Markov chain as defined in 2. □