Bayesian Parameter Inference of Markov Population Model.

Master Thesis

Submitted by

Nhat-Huy Phung

at the

Universität Konstanz

Modeling of Complex, Self-organising Systems

Department of Computer and Information Science

1. **Supervised by:** Jun-.Prof. Dr. Tatjana Petrov
2. **Supervised by:** Prof. Dr. Stefan Leue

Konstanz, 2020

Contents

Acknowledgements

To the completement of this thesis, I would like to describe my deep

Abstract

We present frameworks for data-informed parameter synthesis of Markov population processes. Given statistics data of the population at its steady-state, the object is to synthesize a set of parameters so that a temporal property of interest is satisfied. We design Bayesian frameworks for parameter synthesis in both cases: when the closed form of the interested property is obtainable, and when only simulation is possible. The frameworks are constructed with different sampling and optimization techniques to approximate the posterior distribution. Later, we evaluate the frameworks using different population models of different size using synthetic data generated from a known true parameter. By measuring the distance between synthesized parameters and true parameters and visualize sampled parameter values with their corresponding weights, we show that our frameworks are capable of deriving a set of satisfying parameter values, as well as an estimation which is close to the true parameter.

Chapter 1

Introduction

In different areas of research and application, the objects are to study how the number of individuals in a closed environment develop under a certain set of assumptions. For instance

- Number of online nodes in a distributed system.
- Number of surviving individuals in an epidemic model.

Markov population models [15] are finite state-space, stochastic models that is widely used in modeling complex and dynamic systems. In a Markov population model, each state represents the number of individuals. Formally, in a Markov population model whose state space is $S = (s_1, \dots, s_n)$, there is a map $f : S \to \{0, \dots, N\}$ where $N \in \mathbf{N}^*$ is the maximum number of individuals in the system.
In a Markov population models, for example Discrete-time Markov Chain, initial and transition probabilities are known a-priori. In order to encompass unknown attributes of a system, we introduce *parametric Markov population models*. In a parametric Markov population model, each transition is a rational function of parameters. As unknown features of the system are represented by parameters, the following research questions are raised

- Given a set of data collected by observing the system, how can we know about its parameters?
- Which values of parameters instantiate a model that satisfies a certain property of interest?

Parameter synthesis is an emerging research direction on probabilistic model checking. Katoen [14] define the parameter synthesis problem for pDTMC as to find a set of parameter values, which satisfy a given reachability property. In this thesis, we combines Bayesian parameter inference and parameter synthesis, so that the result parameters (1) satisfy the property of interest, and (2) likely to produce given steady-state data. Contributions of the thesis are

- Presenting and implementing a data-informed, Bayesian frameworks on parameter synthesis of parametric Discrete-time Markov Chain with different case studies.
- Comparing the performances of optimization methods used to approximate posterior distribution in both cases: closed-form solutions are available and only simulations are possible.
- Evaluating the scalability of the frameworks with different sizes of model state-space.

- **Chapter 1** introduces motivations and goals of this research.
- **Chapter 2** presents the theoretical background on probabilistic model checking, include discrete stochastic models and their corresponding temporal logics.
- **Chapter 3** presents essential concepts on Bayesian inference, including sampling and optimization algorithms.
- **Chapter 4** reviews the state-of-the-art works of other researchers on the problem of parameter synthesis.
- **Chapter 5** present Bayesian parameter synthesis frameworks.
- **Chapter 6** describes case studies and benchmarks presented frameworks under different setups.
- **Chapter 7** conclusion and possible future works.

Chapter 2

Probabilistic model checking

We use Discrete-time Markov chain as the formalism to model stochastic population process. In this chapter, we present essential concepts on probabilistic model checking, including probabilistic models and properties. We also briefly present a general deterministic model checking algorithm for a specific temporal logic, namely PCTL. Since it is not always efficient in terms of time complexity to apply deterministic model checking algorithm, we also present a simulation based model checking, namely *statistical model checking* for bounded and unbounded path property which relies only. Since statistical model checking relies only on simulation of stochastic models, it is advantageous for checking models with large space size. We also introduce definitions of parametric model and parameter synthesis problems, as well as the symbolic computing approach to verify parametric models.

2.1 Markov chain

2.1.1 Discrete Time Markov chain

Our definition of markov chain follows the definition on [2].

Definition 2.1.1 (Discrete Time Markov Chain)**.** A Discrete-time Markov chain (DTMC) is a tuple $(S, \mathbf{P}, s_{init}, AP, L)$ where

- S is a countable, non-emty set of *states*

- $\mathbf{P} : S \times S \to [0,1]$ is the *transition probability* function such that

$$\forall s \in S : \sum_{s' \in S} \mathbf{P}(s, s') = 1$$

- $s_{init} : S \to [0,1]$ is the *initial distribution* such that

$$\sum_{s \in S} s_{init}(s) = 1$$

- AP is a set of *atomic propositions*

- $L : S \to 2^{AP}$ is the labelling function on states.

In the scope of this thesis we interest in a special set of states, namely BSCC.

Definition 2.1.2 (Strongly Connected Component)**.** Let $\mathcal{M} = (S, \mathbf{P}, S_{init}, AP, L)$ a DTMC. A subset $S' \subset S$ is strongly connected if and only if for every pair $s_1, s_2 \in S'$ there is a path between s_1 and s_2 which consists of only of state in S'. If there exist no $S'' \subseteq S$, such that $S \subset S''$ and S'' is strongly connected, then S' is a *Strongly Connected Component*, or *SCC* in short.

Definition 2.1.3 (Bottom Strongly Connected Component)**.** Let $\mathcal{M} = (S, \mathbf{P}, S_{init}, AP, L)$ a DTMC and $S' \in S$ a Strongly Connected Component. S' is also a *Bottom Strongly Connected Component*, or *BSCC* for short, if and only if there exist no state $s \in S$
S' that is reachable from any state in S'.

2.1.2 Continuous-time Markov chain

Continous-time Markov chain also satisfies memoryless property

Definition 2.1.4 (Continuous-time Markov property)**.** Let X be a continuous random variable of exponentially distribution. X has memoryless property if and only if

$$Pr\{X > t + \delta | X > t\} = Pr\{X > \delta\} \forall t, \delta \in \mathbb{R}_{\geq 0}$$

The following definition of Contiuous-time Markov chain is based on [3]

Definition 2.1.5 (Continuous-time Markov chain)**.** A Continuous-time Markov chain (CTMC) is a tuple $(S, \mathbf{P}, \mathbf{r}, S_{init}, AP, L)$ [3]

- S is a countable, non-emty set of *states*
- $\mathbf{P} : S \times S \to [0, 1]$ is the *transition probability* function such that

$$\forall s \in S : \sum_{s' \in S} \mathbf{P}(s, s') = 1$$

- $\mathbf{r} : S \to \mathbb{N}$ is the *transition probability* function such that

$$\forall s \in S : \sum_{s' \in S} \mathbf{P}(s, s') = 1$$

- $s_{init} : S \to [0, 1]$ is the *initial distribution* such that

$$\sum_{s \in S} s_{init}(s) = 1$$

- AP is a set of *atomic propositions*
- $L : S \to 2^{AP}$ is the labelling function on states.

2.2 Probabilistic temporal logic

Over CTL properties, we define the set of PCTL properties, in which we ask the probability to have a CTL property satisfied.

Definition 2.2.1 (PCTL* syntax)**.** The syntax of PCTL* is defined as follow

$$\begin{aligned}
&\Phi ::== \text{true} \mid a \mid \Phi \mid \Phi \wedge \Phi \mid \Phi \vee \Phi \mid P_{\sim p}[\phi] \\
&\phi ::== X\Phi \mid \Phi U \Phi
\end{aligned}$$

Definition 2.2.2 (PCTL syntax)**.** The syntax of PCTL is defined as follow

$$\begin{aligned}
&\Phi ::== \text{true} \mid a \mid \Phi \mid \Phi \wedge \Phi \mid \Phi \vee \Phi \mid P_{\sim p}[\phi] \\
&\phi ::== X\Phi \mid \Phi U \Phi
\end{aligned}$$

2.2.1 Model checking PCTL properties

Given a DTMC $\mathcal{M}$ and a PCTL property Φ, general algorithm for checking $\mathcal{M} \models \Phi$ is described as following:

2.2.2 State exlosion problem

2.3 Statistical Model checking

Statistical model checking is a simulation-based approach to model check a statistical model $\mathcal{M}$ against a property Φ

2.3.1 Statistical model checking of unbounded properties.

Estimation method, Chernoff-Hoeffding bound.

2.3.2 Statistical model checking of bounded properties.

We care about

2.4 Parametric model

We introduce parameters to formalize unknown attributes of the system.

Definition 2.4.1 (Polynomial ring)**.** Given a tuple $\mathbf{x} = (x_1, \ldots, x_n)$ be a tuple

Definition 2.4.2. Rational functions Let $\mathbf{x} = \{x_1, \ldots, x_n\}$ be a variable.
Let $\mathbf{Pol}[\mathbf{x}]$ be the set of all polynomial functions over $\mathbf{x}$.
Given $f, g \in \mathbf{Pol}[\mathbf{x}]$, then $h := \frac{f(\mathbf{x})}{g(\mathbf{x})}, g\mathbf{x} \neq 0$ is a rational function over $\mathbf{x}$.
We denote $\mathbb{Q}(\mathbf{x})$ the set of rational functions over $\mathbf{x}$.

2.4.1 Parametric Discrete Time Markov chain

With the set of rational functions formally defined, we define parametric Discrete-time Markov chain based the definition on [13].

Definition 2.4.3 (Discrete Time Markov Chain)**.** A Discrete-time Markov chain (DTMC) is a tuple $(S, \mathbf{x}, \mathbf{P}, s_{init}, AP, L)$ where

- S is a countable, non-emty set of *states*
- $\mathbf{x} \in \mathbb{R}^n, n \in \mathbb{N}$ as the set of n real parameters.
- $\mathbf{P} : S \times S \to \mathbb{Q}(\mathbf{x})$ is the *transition probability* function such that

$$\forall s \in S : \sum_{s' \in S} \mathbf{P}(s, s') = 1$$

- $s_{init} : S \to [0, 1]$ is the *initial distribution* such that

$$\sum_{s \in S} s_{init}(s) = 1$$

- AP is a set of *atomic propositions*
- $L : S \to 2^{AP}$ is the labelling function on states.

Given a parametric Discrete-time Markov chain $\mathcal{M}_\theta$. A concrete assignment of parameter θ *instantiates* a non-parametric Discrete-time Markov chain if $f\theta$ evaluates to a real value for all $f \in \mathbf{P}$.

2.4.2 Symbolic model checking of pDTMC

Example 2.4.1. Parametric Knuth's die We continue the example with Knuth die model $\mathcal{M}_p$. Assume the

$$x =$$

2.4.3 Parameter synthesis of pDTMC

Example 2.4.2. Given a pDTMC of Knuth die $\mathcal{M}_p$ and a path property $\Phi = P_{\geq 0.2}[F\texttt{"one"}]$, synthesize parameter p so that $\mathcal{M}_p \models \Phi$. A simple Monte-Carlo search on parameter space gives the following satisfying point:

2.5 Statistical Model checking

Statistical model checking must be performed on an instantiated Markov chain.

Chapter 3

Bayesian inference

- Bayesian formula: posterior, prior, likelihood
- Bayesian parameter estimation: credible set, Highest density posterior
- Approximation of posterior: tractability and sampling method Monte Carlo (Naive MC, MH, Sequential MC).

3.1 Bayesian inference

3.1.1 Bayesian formula

Let D_{obs} be observed data. In statistical inference, we assume that the observed data has a probability distribution of unknown parameter θ, i.e $D_{obs} \sim P(D_{obs}|\theta)$. In frequentist approach, the estimation of θ based on long-run property, that is, given a large enough sample size, expected value of parameter estimation $\hat{\theta}$ is equal to θ. Therefore, frequentist approach requires to gather a large amount of data to deliver a close estimation $\hat{\theta}$. The main advantage of Bayesian approach over frequentist approach is that it require less data to obtain an estimation $\hat{\theta}$.
In Bayesian approach, we use the information gained from previously observed data *(beliefs)* to enhance the accuracy of the estimation of $\hat{\theta}$. The beliefs obtained from prior knowledge of model parameter θ is represented by *prior distribution* $\pi(\theta)$. We have the *likelihood* $P(D_{obs}|\theta)$ as the probability distribution over observed data, given parameter θ. The Bayesian formula

states that

$$\pi(\theta|D_{obs}) = \frac{P(D_{obs}|\theta)\pi(\theta)}{\int_\theta P(D_{obs}|\theta)\pi(\theta)d\theta}$$

$\int_\theta P(D_{obs}|\theta)\pi(\theta)d\theta$ is the *marginal distribution.* $\pi(\theta|D_{obs})$ is the *posterior distribution.* To compute posterior distribution is the essential part of Bayesian inference, since it gives the estimation of parameter θ.

3.1.2 Bayesian parameter estimation

With posterior distribution $\pi(\theta|D)$ we estimate the parameter $\hat{\theta}$ using Bayesian posterior mean

$$\hat{\theta} = \mathbf{E}[\theta] = \int_\theta \theta\pi(\theta|D)d\theta$$

In case we have samples from posterior distribution, for example the $Trace$ from Metropolis-Hastings algorithm, for example when we use MH algorithm, the discrete form of posterior mean is used:

$$\hat{\theta} = \mathbf{E}[\theta] = \sum_\theta \theta\pi(\theta|D)$$

Definition 3.1.1 (Bayesian Credible Set)**.** Set C is a $(1\alpha)100\%$ credible set for the parameter θ if the posterior probability for θ to belong to C equals (1α).

$$P(\theta \in C|D) = \int_C \pi(\theta|D)d\theta = 1 - \alpha$$

In this thesis, we use by default 0.95 credible set, which corresponds to $\alpha = 0.05$

Definition 3.1.2 (Highest Posterior Density credible set)**.** Highest Posterior Density $(1 - \alpha)100\%$ credible set (HPD for short) is the interval with minimum length over all Bayesian $(1 - \alpha)100\%$ Credible Set.

In this research, the HPD is calculated using algorithm from *PyMC3* library [20]. For simplicity, we assume that in all cases which we concern, HPD is computed for unimodal distribution.

Algorithm 1 Compute Highest Posterior Density Interval

Input: S is samples from a distribution.
Input: $0 \leq \alpha \leq 1$
Output: HPD interval

```
procedure COMPUTE HPD(S)
    Compute interval width w = |S| * α
    Find modal (peak) of sample points.
    Return minimal interval of size |S| − w which contains the modal.
end procedure
```

3.1.3 Selection of prior distribution

Theoretically, prior can be of any distribution family. However, a selection of prior distribution that is too different than the actual distribution of parameter can leads to a false propagation of beliefs and degrade inference results. It is suggested by [19] that in case of no prior knowledge exists to help the selection of prior distribution, Uniform distribution is preferable since it is less likely to propagate false beliefs to the inference.
A systematic inference to select prior distribution family and prior distribution parameter (hyperparameters) is possible with *Hierarchical Bayes Models* [1].

3.1.4 Estimation of posterior distribution

In posterior estimation the following factors are important:

1. Tractability: we have analytical form of posterior distribution.

2. Computationally effective: updating model parameter is of linear time to the dimension of parameter.

Posterior conjugation

Conjugated posteriors are special cases of Bayesian inference, in which the prior and posterior distribution belongs to the same family of distribution. We consider two conjugated posterior: Binomial-Beta and Dirichlet-Multinomial

Lemma 1 (Binomial-Beta Conjugation)**.** Binomial distribution is conjugated to beta distribution.

Proof. The observed data $D = (x_1, \dots, x_n)$ is sampled from $Binomial(k, \theta)$ function

$$P(D|\theta) = \prod_{i=1}^{n} \binom{k}{x_i} \theta^{x_i} (1-\theta)^{k-x_i}$$

The parameter θ is of $Beta(\alpha, \beta)$ distribution

$$\pi(\theta) = \theta^{\alpha-1}(1-\theta)^{\beta-1}$$

We obtained:

$$\begin{aligned}
\pi(\theta|D) &\sim P(D|\theta)\pi(\theta) \\
&\sim \theta^{\sum_{i=1}^{n} x_i}(1-\theta)^{nk-\sum_{i=1}^{n} x_i}\theta^{\alpha-1}(1-\theta)^{\beta-1} \\
&= \theta^{\alpha-1+\sum_{i=1}^{n} x_i}(1-\theta)^{\beta-1+nk-\sum_{i=1}^{n} x_i}
\end{aligned}$$

Thus, the posterior is $Beta(\alpha + \sum_{i=1}^{n} x_i, \beta + nk - \sum_{i=1}^{n} x_i)$ □

Generalize this conjugation, we also have Multinomial-Dirichlet conjugation.

Lemma 2 (Multinomial-Dirichlet Conjugation)**.** Multinomial distribution is conjugated to Dirichlet distribution.

Proof. The observed data $D = (x_1, \dots, x_n)$ is sampled from $Multinomial(n; \theta_1, \dots, \theta_n)$ function

$$P(x_1, \dots, x_n | N, \theta_0, \dots, \theta_n) = \frac{n!}{x_1! \dots x_n!} \prod_{i=1}^{n} \theta_i^{x_i}$$

The parameter $(\theta_1, \dots, \theta_n)$ is $Dirichlet(\alpha_1, \dots, \alpha_n)$

$$\pi(\theta_1, \dots, \theta_n) = \frac{1}{\mathbf{B}(\alpha_1, \dots, \alpha_n)} \prod_{i=1}^{n} \theta_i^{\alpha_i - 1}$$

We obtain

$$\begin{aligned}
\pi(\theta_1, \dots, \theta_n | D) &\sim P(D|\theta)\pi(\theta) \\
&\sim \prod_{i=1}^{n} \theta_i^{x_i} \prod_{i=1}^{n} \theta_i^{\alpha_i - 1} \\
&\sim \prod_{i=1}^{n} \theta_i^{\alpha_i - 1 + \sum_{i=1}^{n} x_i}
\end{aligned}$$

Thus, the posterior is $Dirichlet(\alpha_1 + x_1, \dots, \alpha_n + x_n)$ □

More detailed description in these cases can be found in [22] and [4]. We summarize the necessary results in the following table:

Likelihood	Prior	Posterior parameters
$Binomial(n, k)$	$Beta(\alpha, \beta)$	$\alpha' = \alpha + \sum_{i=1}^{n} x_i$ $\beta' = \beta + nk - \sum_{i=1}^{n} x_i$
$Multinomial(n; \theta_1, \ldots, \theta_n)$	$Dirichlet(\alpha_1, \ldots, \alpha_n)$	$\alpha_i' = \alpha_i + x_i, 1 \leq i \leq n$

However, posterior conjugation is applicable to a subset of prior and likelihood functions. In Bayesian inference, it is usual that the posterior distribution has no analytical form or its analytical form is difficult to directly sample from. In these cases, we can several different sampling and optimization methods to approximate the posterior distribution. In the following section we discuss different approaches for posterior distribution approximation:

- Markov chain Monte-Carlo.
- Sequential Monte-Carlo.
- Approximate Bayesian Computation.

3.1.5 Markov chain Monte-Carlo

In case the posterior distribution has no analytical form or its analytical form is difficult to sample from directly, we use *Metropolis-Hastings* algorithm (*MH* in short).
Metropolis-Hastings algorithm is a *Monte Carlo Markov Chain* algorithm. In its essential, Metropolis-Hastings algorithm draws sample from an unknown distribution. Using the MH algorithm, we can estimate the parameter by posterior mean, without knowing the analytical form of posterior distribution itself.

Algorithm 2 Metropolis-Hastings Algorithm

Input:

- D is the observation data

Output: $Trace$ is the set of accepted sampling point.

procedure METROPOLIS-HASTINGS(D, maxIteration)
  Select a proposal distribution $\pi(\theta)$
  Draw a random initial point θ
  Init empty trace $Trace$
  while maxIteration not reached **do**
    $L \leftarrow P(D|\theta)$
    Draw a point θ' from the proposal distribution.
    $L' \leftarrow P(D|\theta')$
    if $\ln(L') - \ln(L) > 0$ **then**
      Add θ' to $Trace$
      $\theta = \theta'$
    else
      Draw a random number x from $Uniform(0, 1)$
      if $x \leq \xi$, (ξ very small, e.g 10^{-8}) **then**
        Add θ' to $Trace$ (avoiding local maxima)
        $\theta = \theta'$
      end if
    end if
  end while
end procedure

The likelihood function can be implemented as log-likelihood to avoid underflow error. Proposal distribution defines how do we proceed to the next parameter value on the parameter space; it can be of any distribution family.

There are two advantages of using Markov Chain Monte Carlo in Bayesian inference:

1. Parameter transition only needs the computation of likelihood function. Therefore, Monte Carlo Markov Chain can be used in general Bayesian inference, in which we are not guaranteed to have an analytical form of posterior.

2. Specifically in Metropolis-Hastings algorithm, marginal distribution is cancelled out, thus make Metropolis-Hastings a computationally efficient algorithm.

However, MH algorithm also has a drawback; its convergence becomes slower as the dimension of parameter θ increases.

3.1.6 Sequential Monte-Carlo

Sequential Monte-Carlo method is firstly proposed by [6]. Instead of having one particle moving in its parameter space, Sequential Monte-Carlo estimates by using N particles moving independently. Therefore Sequential Monte-Carlo method has a significant advantage of easily parallelizable.

here [5]

Selection of kernel function for SMC is mentioned in [21].

3.1.7 Approximate Bayesian Computation

The methods mentioned before is used with an assumption that the likelihood $P(D_{obs}|\theta)$ has an analytical form; the analytical can be evaluated without introducing computational burden. However there are situations in which the likelihood has no analytical form, or the analytical form is expensive to be evaluated. In such cases, a class of different methods, dubbed *likelihood-free* methods, are used. Likelihood-free methods in Bayesian inference means that instead of compute the likelihood $P(D|\theta)$, we estimate it or replace it by other measures. Approximate Bayesian Computation is a widely used likelihood-free method for approximating posterior distribution. Instead of estimating the likelihood $P(D|\theta)$ directly, we sample a observable data set $\hat{D}$ and define a distance measure $\delta(D, \hat{D})$. Approximate Bayesian Computation accepts a set of tuples $(\hat{\theta}, \hat{D})$, each satisfies that $\delta(D_{obs}, D_{sim}) < \epsilon, \epsilon \in \mathbf{R}_{\leq 0}$.

Algorithm 3 Approximate Bayesian Computation

Input:

- D_{obs}: observed data for Bayesian inference or its summary statistic S_{obs}
- $\theta = (\theta_1, \ldots, \theta_k)$: k-dimensional model parameter.
- $\pi(\theta)$: prior distribution on θ.
- N: number of particles (parameter samples).
- ϵ: absolute error threshold.

Output:

- $(\theta_1, \ldots, \theta_N)$: N sampled particles.
- $(\omega_1, \ldots, \omega_N)$: corresponding weights of sampled particles.

```
procedure APPROXIMATE-BAYESIAN-COMPUTATION(D, θ, π(θ), N, ε)
    t := 0
    while t ≤ N do
    end while
end procedure
```

3.2 Conclusion

We present a set of optimization and approximation methods which are essentials to Bayesian Inference. In the following chapter we propose a data-driven approach for parameter synthesis combining Approximate Bayesian computation, Sequential Monte Carlo, and Statistical Model Checking.

Chapter 4

Related works

[12] [19]

[18] [17]

Polgreen et al [19] presents a method for bayesian inference of pMC parameters in [8]

[16]

The definition and model checking of DTMC and pMC is studied by [2], [11], and [14].

Bayesian inference of pMC parameters is studied in [19] and . In , the authors developed methods to synthesize parameters to satisfy a given set of PCTL properties. In [12], the authors presented methods to perform model checking of biological system using Bayesian statistic. The authors in [12] uses a Bayesian hypothesis test, where H_0 is the null hypothesis that the model satisfies a PCTL P, and alternative hypothesis H_1 is that the system does not satisfies P. Similar approach to the parameter estimation in this project is described by [10].

In this project, we use bee colony model semantics from [9]. The methods and implementation in this project is designed to extend the results of [9] and its tool *DiPS*

storm drawback: it does not support

In [17] the author introduces the same approach but it is to use on CSL properties and CTMC.

Chapter 5

Bayesian frameworks for parameter synthesis.

We present frameworks for data-informed parameter synthesis of pDTMC. The frameworks are designed to synthesize a set of parameter values so that for each value, the instantiated model satisfies the interested property, as

Given a pDTMC model $\mathcal{M}_\theta$, a PCTL property Φ, and observed data D_{obs}, the frameworks synthesize a set of N parameters $(\theta_1, \ldots, \theta_N)$ such that

$$\forall i \in [1, N] : \mathcal{M}_{\theta_i} \models \Phi$$

Eac

5.1 Model checking of parametric models

First way to

Algorithm 4 Markov chain Monte-Carlo with rational functions

Input:

- $\mathcal{M}_\theta$: parametric Discrete-Time Markov chain of parameter θ
- Φ: bounded reachability property of interest.

Output:

- $(\theta_1, \ldots, \theta_{N_{MH}})$: N_{MH} sampled particles.
- $(w_1, \ldots, w_{N_{MH}})$: corresponding weights of sampled particles.

procedure RF-MCMC
end procedure

Algorithm 5 Markov chain Monte-Carlo with rational functions

Input:

- $\mathcal{M}_\theta$: parametric Discrete-Time Markov chain of parameter θ
- Φ: bounded reachability property of interest.

Output:

- $(\theta_1, \ldots, \theta_{N_{MH}})$: N_{MH} sampled particles.
- $(w_1, \ldots, w_{N_{MH}})$: corresponding weights of sampled particles.

procedure RF-MCMC
end procedure

Rational functions are functions of model parameter that represent the probability of finally globally reach each terminal state. The function is delivered by PRISM model checker thanks to it capability of symbolic model checking [**KNP11**].
However, it is not always possible to deduct rational functions from a given model, due to the technical limitation (time, memory) and the limitation of PRISM itself. In our conducted experiment, PRISM is capable of deliver rational functions up to a population of 15 bees. For a population of more

bees, we use the second approach, *DTMC sampling*.
DTMC sampling has advantages over rational function. First, it is less computationally expensive to evaluate a parametric DTMC thanks to its simpler symbolic experession. Second, DTMC sampling is *parallelizable*; sampling can be done with as many processor cores as possible. The second advantage makes DTMC sampling *scalable*, compare to the rational function evaluation approach, which is not scallable due to its nature of deep recursion.

5.2 Bayesian parameter synthesis with rational functions

As we have analytical form for both target property and likelihood function, we can propose a Markov chain Monte-Carlo algorithm. In this case we use Metropolis-Hastings algorithm, with rational function evaluation and model checking is performed before the calculation of acceptance rate.

Algorithm 6 Markov chain Monte-Carlo with rational functions

Input:

- $\mathcal{M}_\theta$: parametric Discrete-Time Markov chain of parameter θ
- Φ: bounded reachability property of interest.
- $\pi(\theta)$: prior distribution on θ.
- N_{MH}: length of particle trace.
- $Q(\theta^t|\theta^{t-1})$: transition kernel.
- D_{obs}: observed data.
- $P(D_{obs}|\theta)$: likelihood function.

Output:

- $(\theta_1, \ldots, \theta_{N_{MH}})$: N_{MH} sampled particles.
- $(w_1, \ldots, w_{N_{MH}})$: corresponding weights of sampled particles.

procedure RF-MCMC
    $sat \leftarrow False$
    while $sat = False$ **do**
        Draw θ_{cand} from $\pi(\theta)$
        Evaluate $val \leftarrow RF_\Phi(\theta)$
        if val satisfies the boundary of Φ **then**
            $sat \leftarrow True$
        end if
    end while
    $\theta_1 \leftarrow \theta_{cand}$
    $w_1 \leftarrow \ln(P(D_{obs}|\theta_{cand}))$

We can also use Sequential Monte-Carlo sampling method.

$i \leftarrow 2$
while $i \leq N_{MH}$ **do**
$sat \leftarrow False$
while $sat = False$ **do**
Draw θ_{cand} from $Q(\theta'|\theta_{i-1})$
Evaluate $val \leftarrow RF_{\Phi}(\theta)$
if val satisfies the boundary of Φ **then**
$sat \leftarrow True$
end if
end while
if $\ln(P(D_{obs}|\theta_{cand})) - \ln(P(D_{obs}|\theta_{i-1})) > 0$ **then**
$\theta_i \leftarrow \theta_{cand}$
$w_i \leftarrow \ln(P(D_{obs}|\theta_{cand}))$
$i \leftarrow i + 1$
else
Draw a random number u from $Uniform(0, 1)$
if $u \leq \xi$, (ξ small, e.g 10^{-2}) **then**
$\theta_i \leftarrow \theta_{cand}$
$w_i \leftarrow \ln(P(D_{obs}|\theta_{cand}))$
$i \leftarrow i + 1$
end if
end if
end while
Return $(\theta_1, \ldots, \theta_{N_{MH}})$, $(w_1, \ldots, w_{N_{MH}})$
end procedure

Algorithm 7 Sequential Monte-Carlo with rational functions

Input:

- $\mathcal{M}_\theta$: parametric Discrete-Time Markov chain of parameter θ
- Φ: bounded reachability property of interest.
- $\pi(\theta)$: prior distribution on θ.
- N: number of particles in the Sequential Monte-Carlo trace.
- M: number of pertubation kernels
- $F_t(\theta^t|\theta_1^{t-1}, \ldots, \theta_N^{t-1}), 1 \leq t \leq M$: pertubation kernels
- N_{MH}: number of particles in each Metropolis-Hastings step.
- $Q_t(\theta^t|\theta^{t-1}), 1 \leq t \leq N_{MH}$: transition kernel for Metropolis-Hastings step.
- D_{obs}: observed data for Bayesian inference or its summary statistic S_{obs}
- $P(D_{obs}|\theta)$: likelihood function.

Output:

- $(\theta_1, \ldots, \theta_N)$: N sampled particles.
- $(w_1, \ldots, w_N)$: corresponding weights of sampled particles.

```
procedure RF-SMC
    i ← 1
    while i ≤ N do                                    ▷ SMC initialization
        Draw θ from π(θ)
        θ_i ← θ
        w_i ← P(D_obs|θ_i)
        i ← i + 1
    end while
```

$t \leftarrow 1$
while $t \leq M$ **do**
  $i \leftarrow 1$ ▷ SMC correction step
  while $i \leq N$ **do**
    $w'_i \leftarrow \frac{w_i}{\sum_{i=1}^{N} w_i}$
  end while
  Sample with replacement $(\theta'_1, \ldots, \theta'_N)$ ▷ SMC selection step
    from $(\theta_1, \ldots, \theta_N)$ with probabilities $(w'_1, \ldots, w'_N)$
  $(\theta_1, \ldots, \theta_N) \leftarrow (\theta'_1, \ldots, \theta'_N)$
  $i \leftarrow 1$
  while $i \leq N$ **do** ▷ SMC pertubation step
    Draw $\hat{\theta}_i^t$ from $F_t(\theta^t | \theta_1^{t-1}, \ldots, \theta_N^{t-1}), 1 \leq t \leq M$
    $(\theta_1^*, \ldots, \theta_{N_{MH}}^*), (w_1^*, \ldots, w_{N_{MH}}^*) \leftarrow RF - MCMC(\hat{\theta}_i^t)$
    $\theta_i \leftarrow \theta_{N_{MH}}^*$
    $w_i \leftarrow w_{N_{MH}}^*$
  end while
end while
Return $(\theta_1, \ldots, \theta_N)$, $(w_1, \ldots, w_N)$
end procedure

5.3 Bayesian frameworks without rational functions

Without the availability of analytical form of observational and interested properties, we face the following obstacles:

- **Absence of likelihood functions:** As the rational functions for properties are not available, we do not have the analytical form of likelihood. The abscence of likelihood suggests to exploit *likelihood-free methods*. In this framework we use *Approximate Bayesian Computation* in combination with *Sequential Monte-Carlo method*.

- **Absence of rational function for verification of bounded path property:** the satisfaction of an instantiated model to a bounded path property cannot be computed. In the case that the number of states is too large, we use *Statistical Model Checking*.

For this case we present Statistical Model Checking, Approximate Bayesian Computation - Sequential Monte-Carlo method *SMC-ABC-SMC* framework.

Algorithm 8 Sequential Monte-Carlo with Approximate Bayesian Computation and Statiscal Model Checking

Input:

- $\mathcal{M}_\theta$: parametric Discrete-Time Markov chain of parameter θ
- Φ: bounded reachability property of interest.
- $\pi(\theta)$: prior distribution on θ.
- N: number of particles in the Sequential Monte-Carlo trace.
- M: number of pertubation kernels
- $F_t(\theta^t|\theta_1^{t-1}, \ldots, \theta_N^{t-1}), 1 \leq t \leq M$
- N_{MH}: number of particles in each Metropolis-Hastings step.
- $Q_t(\theta^t|\theta^{t-1}), 1 \leq t \leq N_{MH}$: transition kernel for Metropolis-Hastings step.
- D_{obs}: observed data for Bayesian inference or its summary statistic S_{obs}
- ϵ: threshold for Approximate Bayesian Computation.
- δ, α: indifference and α-level for Statistical Model Checking using SPRT method.

Output:

- $(\theta_1, \ldots, \theta_N)$: N sampled particles.
- $(w_1, \ldots, w_N)$: corresponding weights of sampled particles.

procedure SMC-ABC-SMC
    $i \leftarrow 1$
    while $i \leq N$ **do** ▷ SMC initialization
        Draw θ from $\pi(\theta)$
        $\theta_i \leftarrow \theta$
        $w_i \leftarrow 1$
    end while

$t \leftarrow 1$
while $t \leq M$ **do**
  $i \leftarrow 1$ ▷ SMC correction step
  while $i \leq N$ **do**
    $w'_i \leftarrow \frac{w_i}{\sum_{i=1}^{N} w_i}$
  end while
  Sample with replacement $(\theta'_1, \ldots, \theta'_N)$ ▷ SMC selection step
   from $(\theta_1, \ldots, \theta_N)$ with probabilities $(w'_1, \ldots, w'_N)$
  $(\theta_1, \ldots, \theta_N) \leftarrow (\theta'_1, \ldots, \theta'_N)$
  $i \leftarrow 1$
  while $i \leq N$ **do** ▷ SMC pertubation step
    $rejected \leftarrow True$
    while $rejected == True$ **do**
      $sat \leftarrow False$
      while $sat = False$ **do**
        Draw $\hat{\theta}_i^t$ from $F_t(\theta^t | \theta_1^{t-1}, \ldots, \theta_N^{t-1}), 1 \leq t \leq M$
        Do SPRT SMC on $\mathcal{M}_{\hat{\theta}^t}$ and Φ
        if $\mathcal{M}_{\hat{\theta}^t} \models \Phi$ **then**
          $sat \leftarrow True$
        end if
      end while
      $D_{sim} \leftarrow Simulate(\mathcal{M}_{\hat{\theta}^t})$
      $d = Distance(D_{sim}, D_{obs})$
      if $d < \epsilon$ **then**
        $rejected \leftarrow False$
        $\theta_i \leftarrow \hat{\theta}^t$
        $w_i \leftarrow d$
      end if
    end while
  end while
end while
Return $(\theta_1, \ldots, \theta_N)$, $(w_1, \ldots, w_N)$
end procedure

5.4 Selection of pertubation kernel

Selection of pertubation kernel is mentioned in [7]. In this thesis, we use component-wise uniform kernel:

Chapter 6

Case study

6.1 Zeroconf

6.1.1 System description

Zero-configuration protocol (*zeroconf* for short) is a protocol used in IPv4 network to allocate newly attached device an unique IP address without any intervention from network operators.

6.1.2 Model and properties

From the pseudocode of Zeroconf protocol

6.1.3 Evaluation

6.1.4 Conclusion

6.2 Bees colony

6.2.1 System description

We study the collective behavior of a bee colony. Each bee in a colony possibly stings after observing a threat in the surrounding environment, and warn other bees by releasing a special substance, pheromone. By sensing the pheromone released in the environment, other bees in the colony may also sting. However, since stinging leads to the termination of an individual bee, it reduces the total defense capability as well. With parametric Discrete-time Markov chain as the model, we study how the actions of a single bee change with regarding to the colony size of and pheromone amount.

6.2.2 Model and properties

Assume that each bee in a colony decides its next action (to sting or not to sting) based only on the current state of the environment, and the number of bees who sting or not sting can be modeled as a Markov process. To reduce the complexity of the model, we make another assumption that the states of the bees colony are observed after uniform time duration, hence the model is of discrete-time.There are 3 assumptions on the system:

1. Each bee release an unit amount of pheromone immediately after stinging.
2. A bee dies after stinging and releasing pheromone. In the other words, no bee can sting more than once.
3. Stinging behaviour only depends on the concentration of pheromone in the environment.

Under these assumption, a bee colony can be viewed as a set of agents (bees) interact with each other in a closed environment with the appearance of a factor *pheromone*. Afterward, the agent has probability to commit an action, namely *sting*. The agent is eliminated from environment after stinging. Assume that we have a colony of n bees initially. As aforementioned, an individual bee is terminated after it stings. Thus, at the end of experiment, the number of bees is $n' \in \{0, 1, \ldots, n\}$. We model the bee colony with a DTMC $\mathcal{M} = (S, \mathbf{P}, S_{init}, AP, L)$, such that

- $|S_{init}| = 1$
- There exists $n + 1$ tSCCs which encode the population at the end of the experiment.

Semantics of Markov population models for bees colony are developed by [9].

6.2.3 Evaluation

6.2.4 Conclusion

6.3 SIR model

6.3.1 System

SIR model is a population model, which is widely used in modeling epidemics. In a SIR model, each individual is of one among three types:

- *Susceptible (S)*
- *Infected (S)*
- *Recovered (S)*

$$S + I \xrightarrow{\alpha} 2I$$
$$I \xrightarrow{\beta} R$$

6.3.2 Model and properties

Example of an SIR CTMC model with initial population $(S_0, I_0, R_0) = (3, 1, 0)$

Uniformize the chain with uniformization rate $(3\alpha + 4\beta)$, we derive the following uniformized DTMC:

6.3.3 Properties

6.3.4 Evaluation

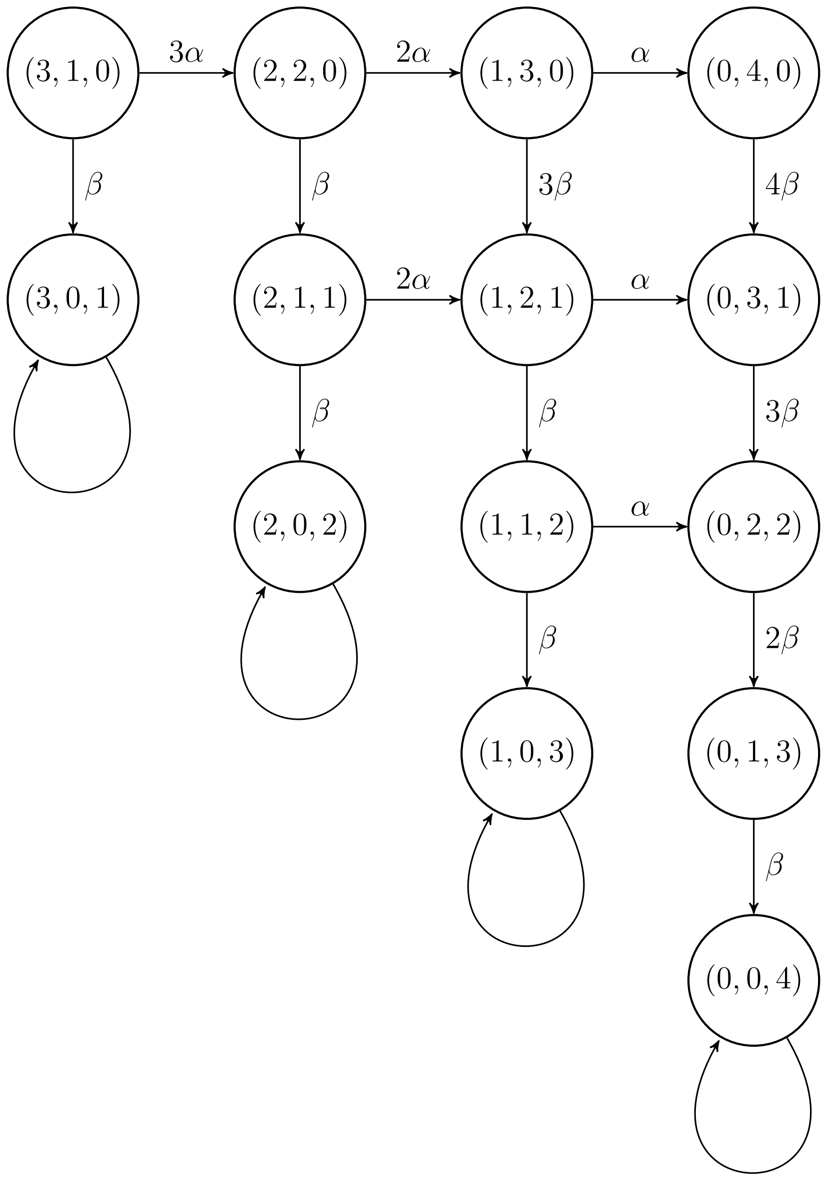


Figure 6.1: $SIR(3,1,0)$ CTMC model with parameters(α, β)

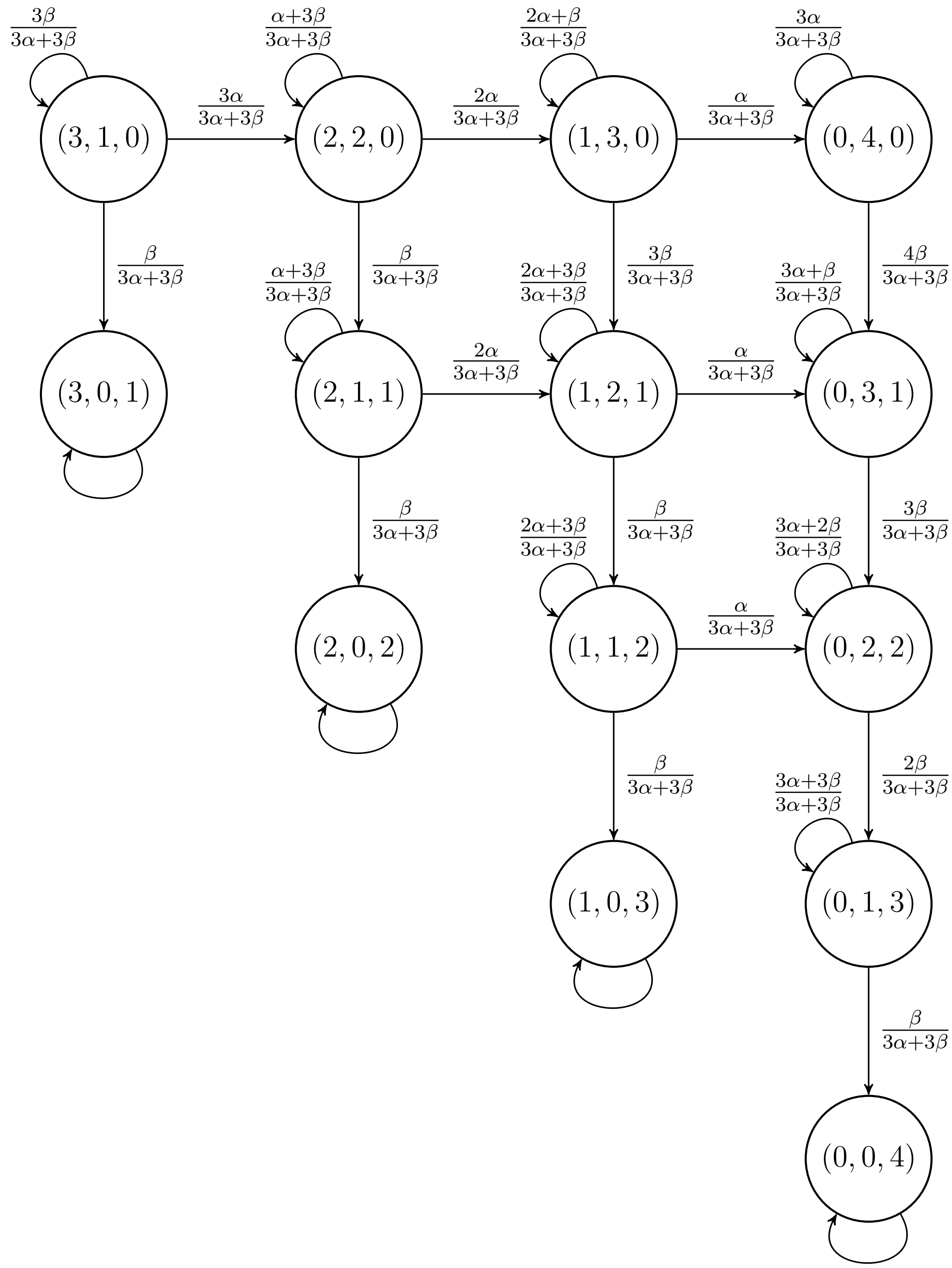


Figure 6.2: $SIR(3,1,0)$ Uniformized DTMC model with parameters(α,β) and uniformization rate $(3\alpha+4\beta)$

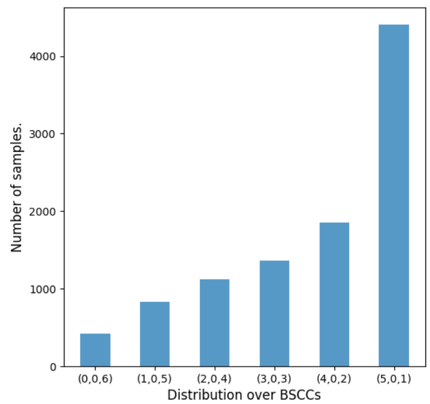
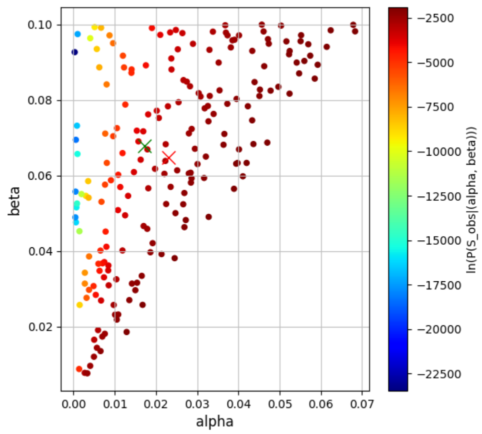


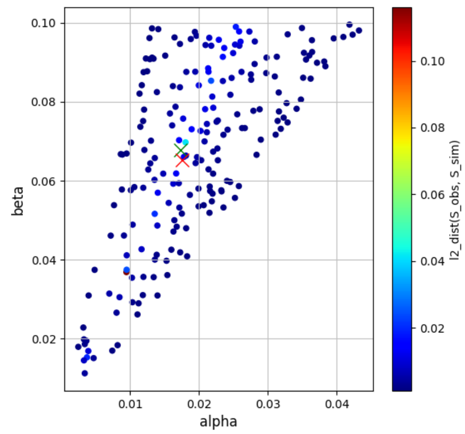
Figure 6.3: Synthetic data y_{obs} using selected true parameter.

SIR(5,1,0)	Rational function SMC	Statistical model checking ABC-SMC
True parameter	(0.01724649, 0.06778604)	
Number of states	27	
Number of BSCCs	6	
Target property	$P_{\geq 0.25}[!(i > 2)U^{<6}(i = 0)]$	
Synthetic data	(421, 834, 1126, 1362, 1851, 4406)	
Inferred parameter point	(0.02307652, 0.06481155)	(0.01758384, 0.06535699)
L2 distance to true parameter	0.006544985909916083	0.005519695496673707
Run time (hh:mm:ss)	1:07:36.442146	3:05:22.61795

Table 6.1: SIR(5,1,0) parameter estimation results.



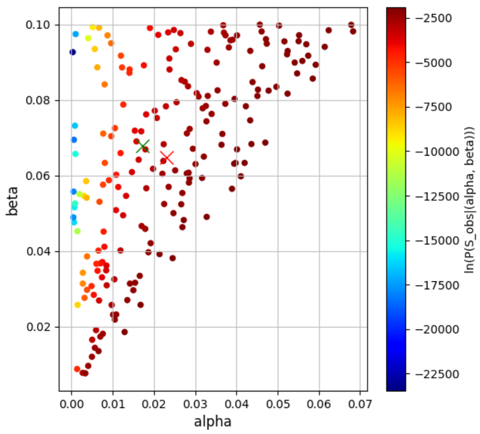
(a) Sampled particles using Rational Functions SMC



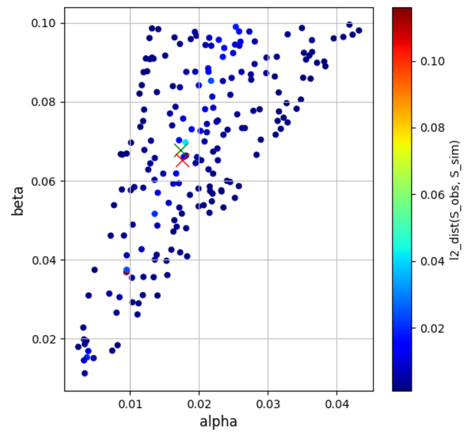
(b) Sampled particles using Statiscal Model Checking ABC-SMC

SIR(10,1,0)	Rational function SMC	Statistical model checking ABC-SMC
True parameter	(0.01724649, 0.06778604)	
Number of states	27	
Number of BSCCs	6	
Target property	$P_{\geq 0.25}[!(i > 2)U^{<6}(i = 0)]$	
Synthetic data	(421, 834, 1126, 1362, 1851, 4406)	
Inferred parameter point	(0.02307652, 0.06481155)	(0.01758384, 0.06535699)
L2 distance to true parameter	0.006544985909916083	0.005519695496673707
Run time (hh:mm:ss)	1:07:36.442146	3:05:22.61795

Table 6.2: SIR(5,1,0) parameter estimation results.



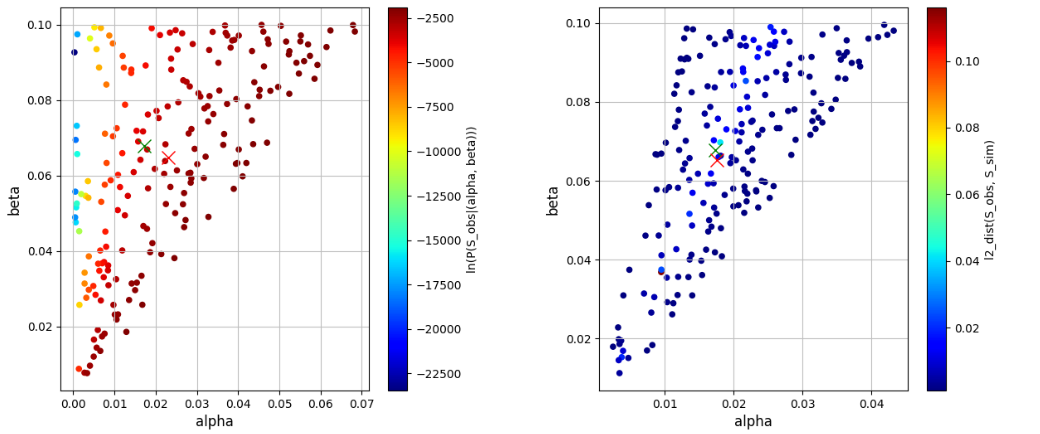
(a) Sampled particles using Rational Functions SMC



(b) Sampled particles using Statiscal Model Checking ABC-SMC

SIR(15,1,0)	Rational function SMC	Statistical model checking ABC-SMC
True parameter	(0.01724649, 0.06778604)	
Number of states	27	
Number of BSCCs	6	
Target property	$P_{\geq 0.25}[!(i > 2)U^{<6}(i = 0)]$	
Synthetic data	(421, 834, 1126, 1362, 1851, 4406)	
Inferred parameter point	(0.02307652, 0.06481155)	(0.01758384, 0.06535699)
L2 distance to true parameter	0.006544985909916083	0.005519695496673707
Run time (hh:mm:ss)	1:07:36.442146	3:05:22.61795

Table 6.3: SIR(5,1,0) parameter estimation results.

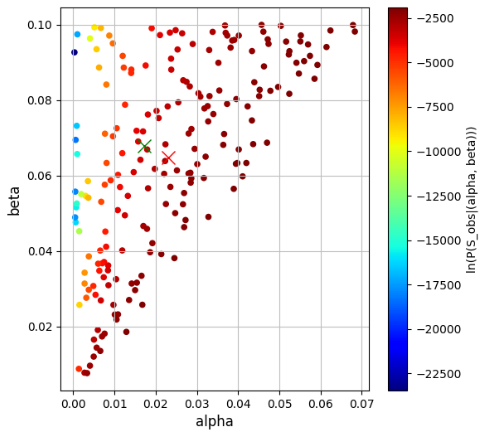


(a) Sampled particles using Rational Functions SMC

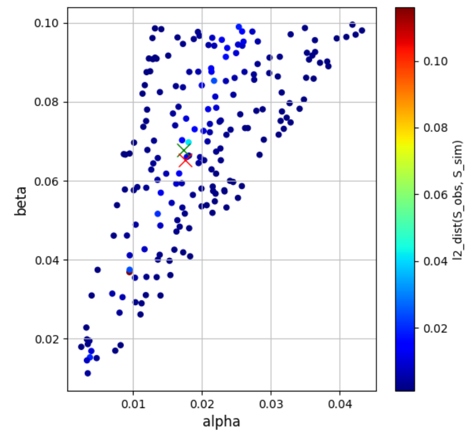
(b) Sampled particles using Statiscal Model Checking ABC-SMC

SIR(10,1,0), BSCC merged	Rational function SMC	Statistical model checking ABC-SMC
True parameter	(0.01724649, 0.06778604)	
Number of states	27	
Number of BSCCs	6	
Target property	$P_{\geq 0.25}[!(i > 2)U^{<6}(i = 0)]$	
Synthetic data	(421, 834, 1126, 1362, 1851, 4406)	
Inferred parameter point	(0.02307652, 0.06481155)	(0.01758384, 0.06535699)
L2 distance to true parameter	0.006544985909916083	0.005519695496673707
Run time (hh:mm:ss)	1:07:36.442146	3:05:22.61795

Table 6.4: SIR(5,1,0) parameter estimation results.



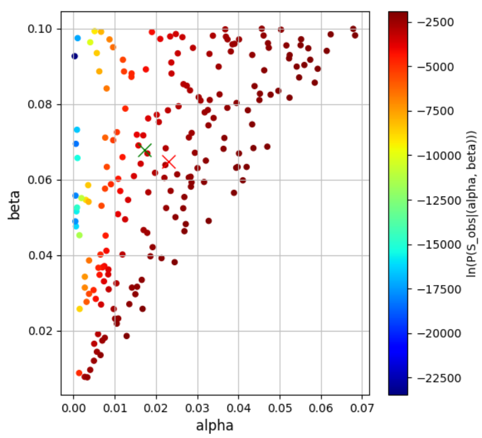
(a) Sampled particles using Rational Functions SMC



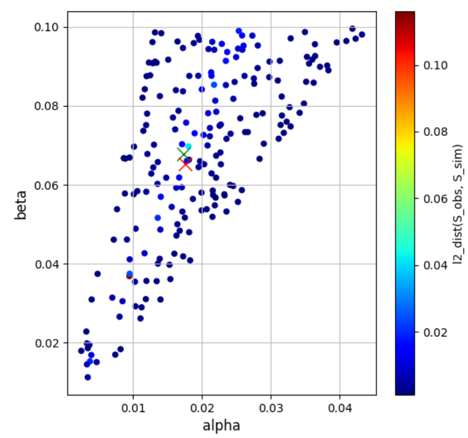
(b) Sampled particles using Statiscal Model Checking ABC-SMC

SIR(10,1,0), BSCC merged	Rational function SMC	Statistical model checking ABC-SMC
True parameter	(0.01724649, 0.06778604)	
Number of states	27	
Number of BSCCs	6	
Target property	$P_{\geq 0.25}[!(i > 2)U^{<6}(i = 0)]$	
Synthetic data	(421, 834, 1126, 1362, 1851, 4406)	
Inferred parameter point	(0.02307652, 0.06481155)	(0.01758384, 0.06535699)
L2 distance to true parameter	0.006544985909916083	0.005519695496673707
Run time (hh:mm:ss)	1:07:36.442146	3:05:22.61795

Table 6.5: SIR(5,1,0) parameter estimation results.



(a) Sampled particles using Rational Functions SMC



(b) Sampled particles using Statiscal Model Checking ABC-SMC

Chapter 7

Conclusion

7.1 Summary

In this thesis we shows the possibility to infer the parameters of

7.2 Future works

Bibliography

[1] Greg M Allenby, Peter E Rossi, and RE McCulloch. "Hierarchical Bayes Models: A Practitioners Guide. Grover R, Vriens M, eds". In: *SSRN Electron J* (2005).

[2] Christel Baier and Joost-Pieter Katoen. *Principles of model checking*. MIT press, 2008.

[3] Christel Baier et al. "Model-checking algorithms for continuous-time Markov chains". In: *IEEE Transactions on software engineering* 29.6 (2003), pp. 524–541.

[4] Michael Baron. *Probability and statistics for computer scientists*. CRC Press, 2019.

[5] Remi Daviet. "Inference with Hamiltonian Sequential Monte Carlo Simulators". In: *arXiv preprint arXiv:1812.07978* (2018).

[6] Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. "Sequential monte carlo samplers". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 68.3 (2006), pp. 411–436.

[7] Sarah Filippi et al. "On optimality of kernels for approximate Bayesian computation using sequential Monte Carlo". In: *Statistical applications in genetics and molecular biology* 12.1 (2013), pp. 87–107.

[8] Sofie Haesaert, Alessandro Abate, and Paul MJ Van den Hof. "Data-driven and model-based verification: A bayesian identification approach". In: *2015 54th IEEE Conference on Decision and Control (CDC)*. IEEE. 2015, pp. 6830–6835.

[9] Matej Hajnal et al. "Data-Informed Parameter Synthesis for Population Markov Chains". In: *International Workshop on Hybrid Systems Biology*. Springer. 2019, pp. 147–164.

[10] Faraz Hussain et al. "Automated parameter estimation for biological models using Bayesian statistical model checking". In: *BMC bioinformatics* 16.S17 (2015), S8.

[11] Lisa Hutschenreiter, Christel Baier, and Joachim Klein. "Parametric Markov chains: PCTL complexity and fraction-free Gaussian elimination". In: *arXiv preprint arXiv:1709.02093* (2017).

[12] Sumit K Jha et al. "A bayesian approach to model checking biological systems". In: *International conference on computational methods in systems biology*. Springer. 2009, pp. 218–234.

[13] Sebastian Junges et al. "Parameter synthesis for Markov models". In: *arXiv preprint arXiv:1903.07993* (2019).

[14] Joost-Pieter Katoen. "The probabilistic model checking landscape". In: *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science*. 2016, pp. 31–45.

[15] John FC Kingman. "Markov population processes". In: *Journal of Applied Probability* (1969), pp. 1–18.

[16] Marta Kwiatkowska, Gethin Norman, and David Parker. "PRISM 4.0: Verification of probabilistic real-time systems". In: *International conference on computer aided verification*. Springer. 2011, pp. 585–591.

[17] Gareth W Molyneux and Alessandro Abate. "ABC(SMC)^2: Simultaneous Inference and Model Checking of Chemical Reaction Networks". In: *International Conference on Computational Methods in Systems Biology*. Springer. 2020, pp. 255–279.

[18] Gareth W Molyneux, Viraj B Wijesuriya, and Alessandro Abate. "Bayesian verification of chemical reaction networks". In: *International Symposium on Formal Methods*. Springer. 2019, pp. 461–479.

[19] Elizabeth Polgreen et al. "Data-efficient Bayesian verification of parametric Markov chains". In: *International Conference on Quantitative Evaluation of Systems*. Springer. 2016, pp. 35–51.

[20] John Salvatier, Thomas V Wieckiâ, and Christopher Fonnesbeck. "PyMC3: Python probabilistic programming framework". In: *ascl* (2016), ascl–1610.

[21] Daniel Silk, Saran Filippi, and Michael PH Stumpf. “Optimizing threshold-schedules for approximate Bayesian computation sequential Monte Carlo samplers: applications to molecular systems”. In: *arXiv preprint arXiv:1210.3296* (2012).

[22] Stephen Tu. “The dirichlet-multinomial and dirichlet-categorical models for bayesian inference”. In: *Computer Science Division, UC Berkeley* (2014).