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

something

Chapter 1

Introduction

1.1 Motivation

In different areas of research and application, the objects are to investigate how the number of individuals changes 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 [14] 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 $S = (s_1, \ldots, s_n)$, there is a map $f : S \to \{0, 1, \ldots, N-1, N\}$ where $N \in \mathbf{N}^*$ is the maximum number of individuals in the system.
In a Markov population models, such as Discrete-time Markov Chain, initial and transition probabilities are known a-priori. In order to formalize unknown attributes of a system, we introduce *parametric Markov population models.* In a parametric Markov population model, each transition is a 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 warrant that a certain property holds on our model?

In this thesis, we work with a specific parametric Markov population model, that is *parametric Discrete-time Markov Chain.* In order to answer the aforementioned research questions, we presents a data-driven approach for parameter synthesis of parametric Discrete-time Markov Chain.Parameter synthesis is an emerging research direction on probabilistic model checking. Parameter synthesis problem is to find a set of parameter values to satisfy a certain reachability property [13].

1.2 Contribution

Contributions of this thesis are

- Investigate a data-driven approach on parameter synthesis of parametric Discrete-time Markov Chain.
- Evaluate the scalability of the approach in cases of closed-form solution available and simulation.
- Compare the performances of optimization methods used to approximate posterior distribution.

1.3 Structure of the thesis

- **Chapter 1** introduces motivations for the research topic.
- **Chapter 2** presents the theoretical background on probabilistic model checking, include discrete stochastic models and their corresponding temporal logics.
- **Chapter 3**
- **Chapter 4** reviews the state-of-the-art works of other researchers on the problem of parameter synthesis.
- **Chapter 5** describes the benchmark.
- **Chapter 6** conclusion and future work.

Chapter 2

Probabilistic model checking

- Discrete time Markov chain
- Continuous time Markov chain, conversion to discrete time chain
- Probabilistic Temporal logics
- Probabilistic model checking
- Statistical model checking
- Parametric Discre time Markov chain
- Parameter synthesis problem: approaches: bayesian, Abate paper on bayesian

In this thesis, we model stochastic systems. Thus, we use probabilistic models, in our case we mostly use Discrete time markov chain, and also have a

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.

2.1.2 Continuous-time Markov chain

Continous-time Markov chain also satisfies memoryless property

Definition 2.1.2 (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 2.1.3 (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

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

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

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

- AP is a set of *atomic propositions*
- $L : S \rightarrow 2^{AP}$ 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

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

2.3 Model checking PCTL properties

2.4 Parametric model

We introduce parameters to formalize unknown attributes of the system.

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

Definition 2.4.2. Rational functions Let $\mathbf{x} = \{x_1, \dots, x_n\}$ 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 [12].

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.
- $\mathbf{P} : S \times S \rightarrow \mathbb{Q}(\mathbf{x})$ is the *transition probability* function such that

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

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

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

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

Given a parametric Discrete-time Markov chain M_p. A concrete assignment of parameter $\mathbf{x}$ *instantiate* a non-parametric Discrete-time Markov chain if $f\mathbf{x}$ evaluates to a real value for all $f \in \mathbf{P}$.

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 be observed data. In statistical inference, we assume that the observed data has a probability distribution of unknown parameter θ, i.e $D \sim P(D|\theta)$. 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 $\hat{\theta}$. In Bayesian approach, we reuse the information *beliefs* gained from observed data to enhance the accuracy of the estimation of $\hat{\theta}$. The main advantage of Bayesian approach over frequentist approach is that it require less data to obtain an estimation $\hat{\theta}$.
The beliefs obtained from prior knowledge of model parameter θ is represented by *prior distribution* $\pi(\theta)$.
Also, we have probability distribution of observed data, given parameter θ, $P(D|\theta)$. This is also called *likelihood function*.

With Bayesian formula, we have

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

$\int_\theta P(D|\theta)\pi(\theta)d\theta$ is called *marginal distribution*. $\pi(\theta|D)$ is called *posterior distribution*. Computing posterior distribution is the essential part of Bayesian inference, since it gives us 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 [17]. 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 $w = |S| * \alpha$
    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 [16] 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 [19] 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 $\theta$
    Init empty trace $Trace$
    while maxIteration not reached do
        $L \leftarrow P(D|\theta)$
        Draw a point $\theta'$ from the proposal distribution.
        $L' \leftarrow P(D|\theta')$
        if $\ln(L') - \ln(L) > 0$ then
            Add $\theta'$ to $Trace$
            $\theta = \theta'$
        else
            Draw a random number $x$ from $Uniform(0, 1)$
            if $x \leq \xi$, ($\xi$ very small, e.g $10^{-8}$) then
                Add $\theta'$ 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 [18].

3.1.7 Approximate Bayesian computation

The methods mentioned before is used with an assumption that the likelihood $P(D|\theta)$ 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 $\delta(D, \hat{D}) < \epsilon, \epsilon \in \mathbf{R}_{\leq 0}$.

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($D$, $\theta$, $\pi(\theta)$, $N$, $\epsilon$)
    $t := 0$
    while $t \leq N$ 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

The current research progress on probabilistic model checking is studied thoroughly by Katoen and Baier et al [2]. Katoen et al. [13] briefly summarized important aspect of probabilistic model checking.

Polgreen et al [16] presents a method for bayesian inference of pMC parameters in

The definition and model checking of DTMC and pMC is studied by [2], [10], and [13].
Bayesian inference of pMC parameters is studied in [16] and [11]. In [16], the authors developed methods to synthesize parameters to satisfy a given set of PCTL properties. In [11], the authors presented methods to perform model checking of biological system using Bayesian statistic. The authors in [11] 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 [9].
In this project, we use bee colony model semantics from [8]. The methods and implementation in this project is designed to extend the results of [8] and its tool *DiPS*

storm drawback: it does not support

In [15] the author introduces the same approach but it is to use on CSL properties and CTMC.

Chapter 5

Bayesian frameworks for parameter synthesis.

5.1 Bayesian frameworks 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 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.
- $RF_\Phi(\theta)$: rational function of target property.
- $\pi(\theta)$: 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 $\theta_{cand}$ from $\pi(\theta)$
        Evaluate $val \leftarrow RF_\Phi(\theta)$
        if $val$ satisfies the boundary of $\Phi$ then
            $sat \leftarrow True$
        end if
    end while
    $\theta_1 \leftarrow \theta_{cand}$
    $w_1 \leftarrow \ln(P(D_{obs}|\theta_{cand}))$
    $i \leftarrow 2$
    **while** $i \leq N_{MH}$ **do**
        $sat \leftarrow False$
        **while** $sat = False$ **do**
            Draw $\theta_{cand}$ from $Q(\theta'|\theta_{i-1})$
            Evaluate $val \leftarrow RF_{\Phi}(\theta)$
            **if** $val$ satisfies the boundary of $\Phi$ **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** $u \leq \xi$, ($\xi$ 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**
```

We can also use Sequential Monte-Carlo sampling method.

Algorithm 5 Sequential Monte-Carlo with rational functions

Input:

- $\mathcal{M}_\theta$: parametric Discrete-Time Markov chain of parameter θ
- Φ: bounded reachability property of interest.
- $RF_\Phi(\theta)$: rational function of target property.
- $\pi(\theta)$: 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
    $i \leftarrow 1$
    while $i \leq N$ do                                  ▷ SMC initialization
        Draw $\theta$ from $\pi(\theta)$
        $\theta_i \leftarrow \theta$
        $w_i \leftarrow P(D_{obs}|\theta_i)$
    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
```

5.2 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 6 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 $\theta$ from $\pi(\theta)$
        $\theta_i \leftarrow \theta$
        $w_i \leftarrow P(D_{obs}|\theta_i)$
    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 $\hat{\theta}^t_i$ from $F_t(\theta^t | \theta^{t-1}_1, \ldots, \theta^{t-1}_N), 1 \leq t \leq M$
                    Do SPRT SMC on $\mathcal{M}_{\hat{\theta}^t}$ and $\Phi$
                    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
```

5.3 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 [8].

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)*

$$\begin{aligned} S + I &\xrightarrow{\alpha} 2I \\ I &\xrightarrow{\beta} R \end{aligned}$$

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

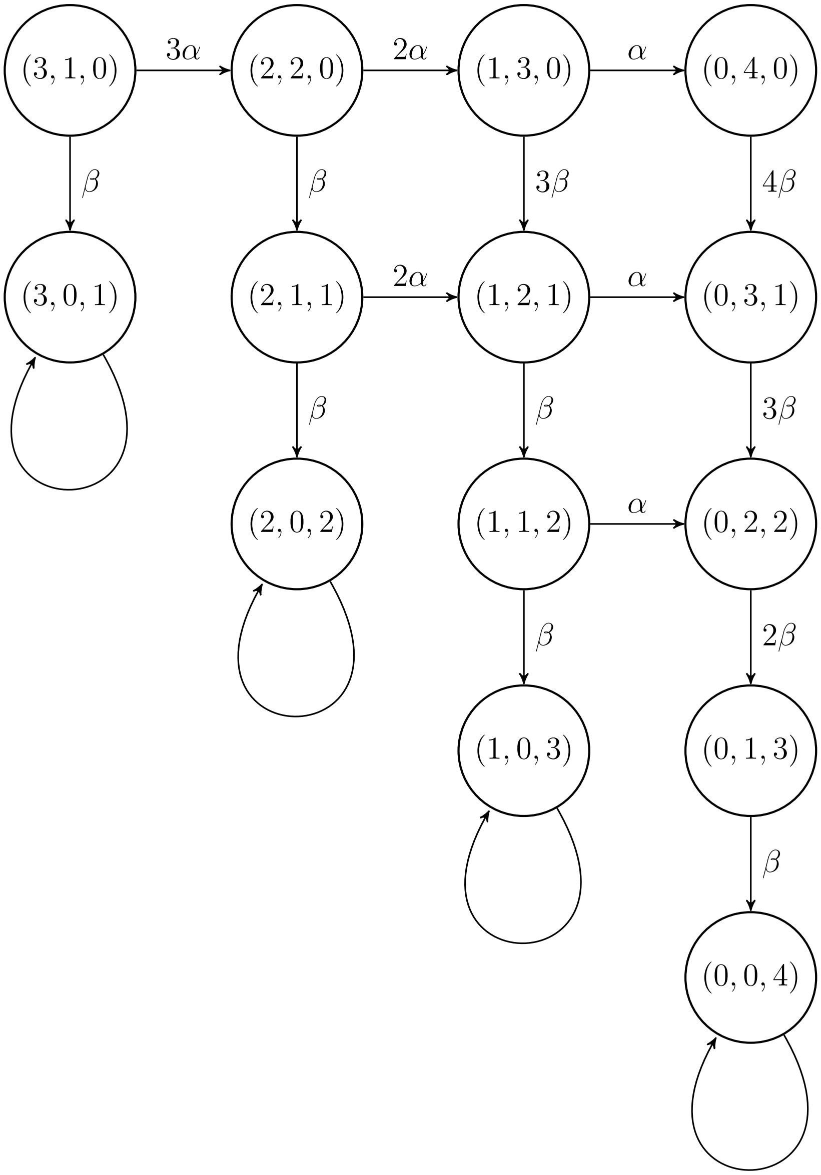


Figure 6.1: $SIR(3,1,0)$ CTMC model with parameters(α,β)

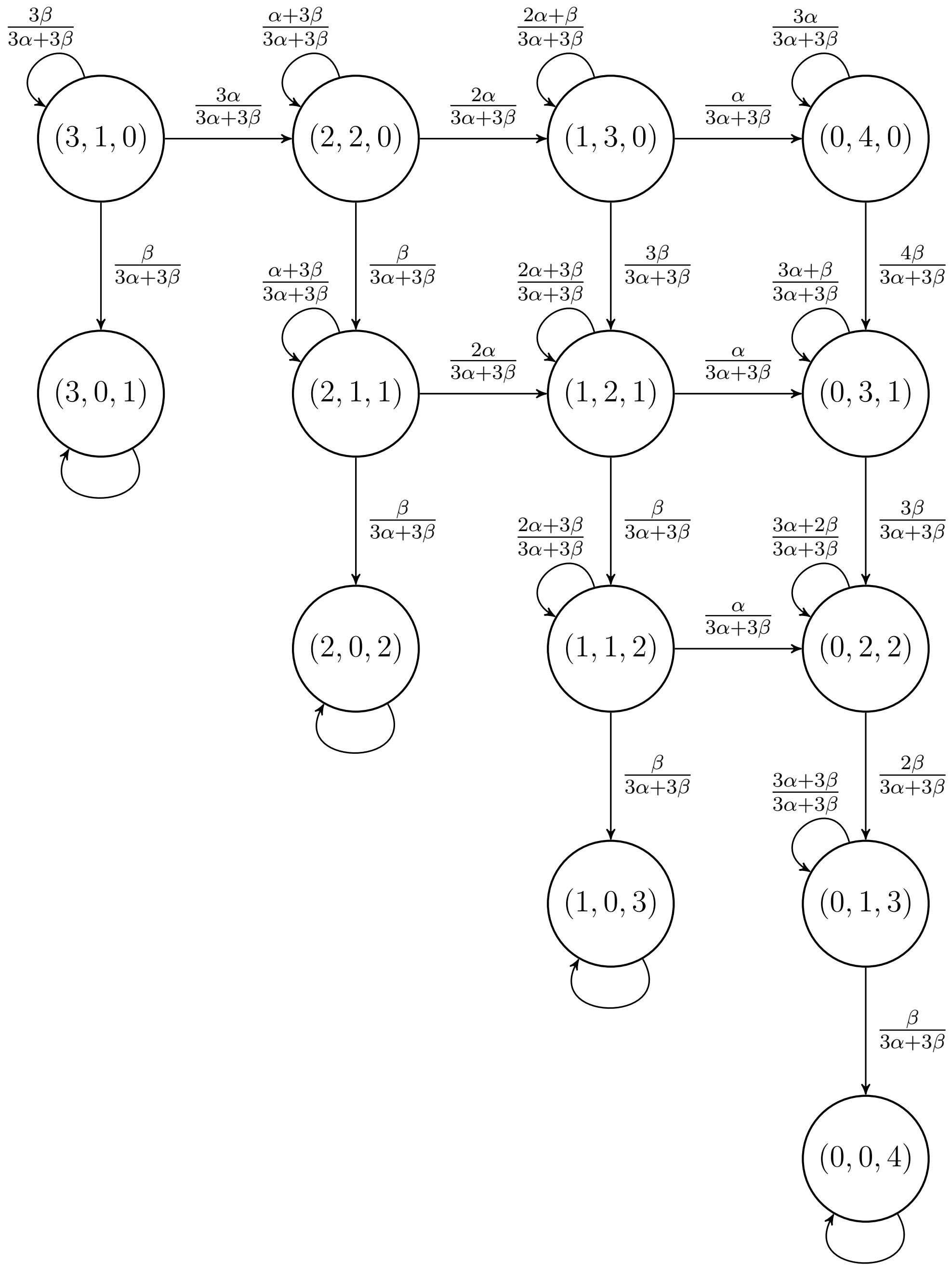

$\frac{3\beta}{3\alpha+3\beta}$
$\frac{\alpha+3\beta}{3\alpha+3\beta}$
$\frac{2\alpha+\beta}{3\alpha+3\beta}$
$\frac{3\alpha}{3\alpha+3\beta}$
$\frac{3\alpha}{3\alpha+3\beta}$
$\frac{2\alpha}{3\alpha+3\beta}$
$\frac{\alpha}{3\alpha+3\beta}$
$(3,1,0)$
$(2,2,0)$
$(1,3,0)$
$(0,4,0)$
$\frac{\beta}{3\alpha+3\beta}$
$\frac{\alpha+3\beta}{3\alpha+3\beta}$
$\frac{\beta}{3\alpha+3\beta}$
$\frac{2\alpha+3\beta}{3\alpha+3\beta}$
$\frac{3\beta}{3\alpha+3\beta}$
$\frac{3\alpha+\beta}{3\alpha+3\beta}$
$\frac{4\beta}{3\alpha+3\beta}$
$\frac{2\alpha}{3\alpha+3\beta}$
$\frac{\alpha}{3\alpha+3\beta}$
$(3,0,1)$
$(2,1,1)$
$(1,2,1)$
$(0,3,1)$
$\frac{\beta}{3\alpha+3\beta}$
$\frac{2\alpha+3\beta}{3\alpha+3\beta}$
$\frac{\beta}{3\alpha+3\beta}$
$\frac{3\alpha+2\beta}{3\alpha+3\beta}$
$\frac{3\beta}{3\alpha+3\beta}$
$\frac{\alpha}{3\alpha+3\beta}$
$(2,0,2)$
$(1,1,2)$
$(0,2,2)$
$\frac{\beta}{3\alpha+3\beta}$
$\frac{3\alpha+3\beta}{3\alpha+3\beta}$
$\frac{2\beta}{3\alpha+3\beta}$
$(1,0,3)$
$(0,1,3)$
$\frac{\beta}{3\alpha+3\beta}$
$(0,0,4)$

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

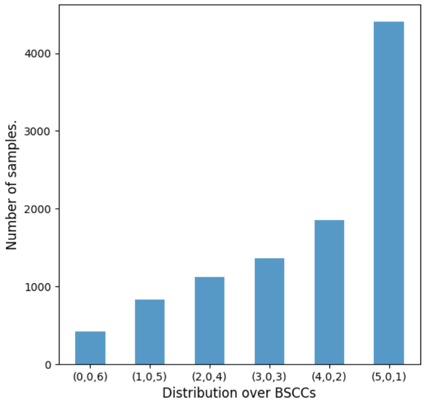
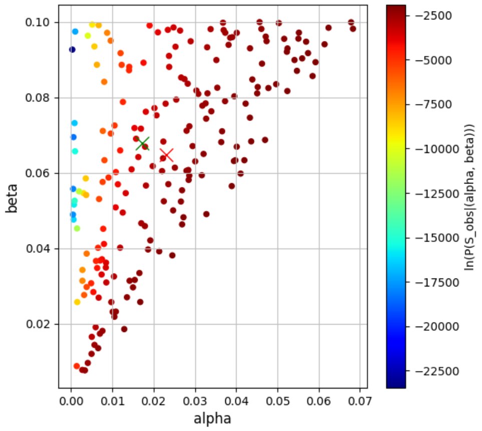


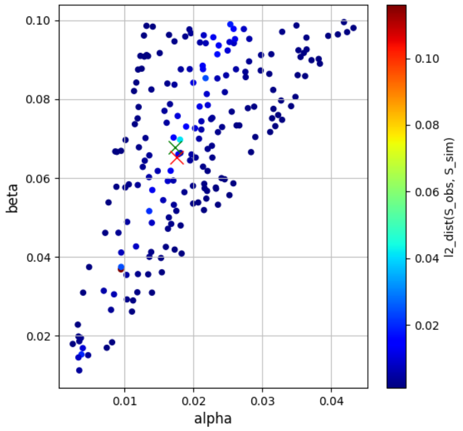
Figure 6.3: Synthetic data y_{obs} using selected true parameter.

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.



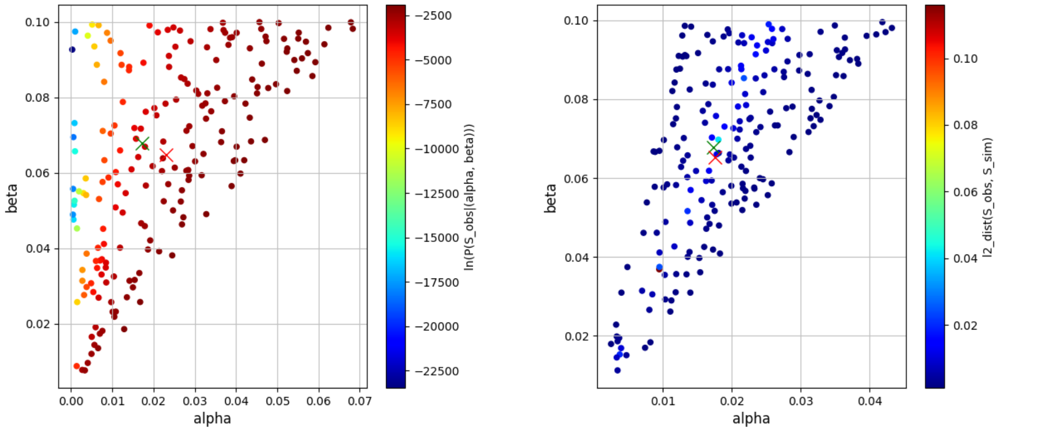
(a) Sampled particles using Rational Functions SMC



(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.

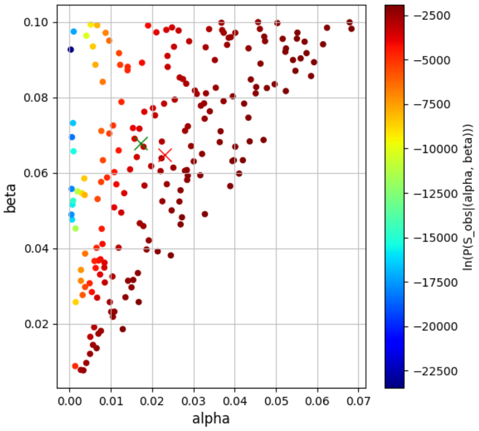


(a) Sampled particles using Rational Functions SMC

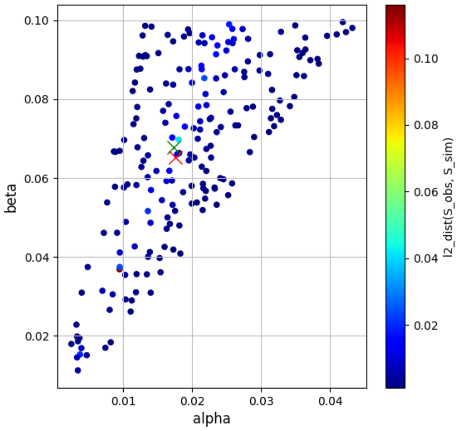
(b) Sampled particles using Statiscal Model Checking ABC-SMC

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.



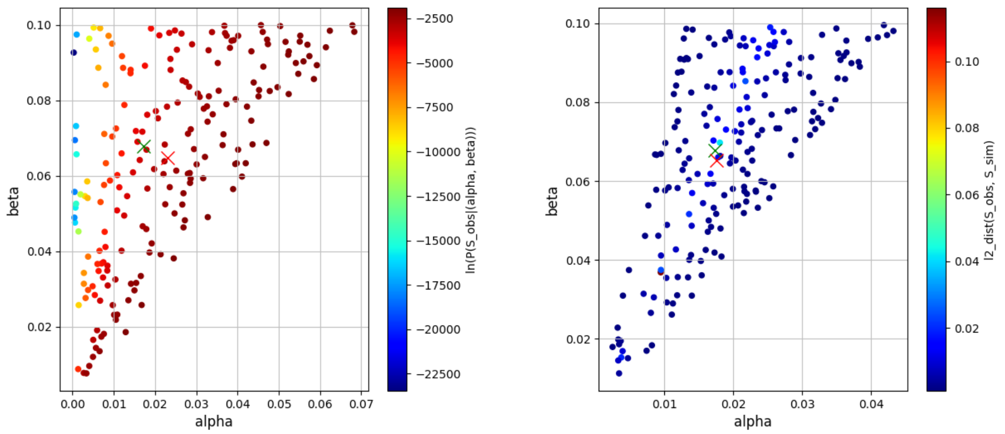
(a) Sampled particles using Rational Functions SMC



(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.

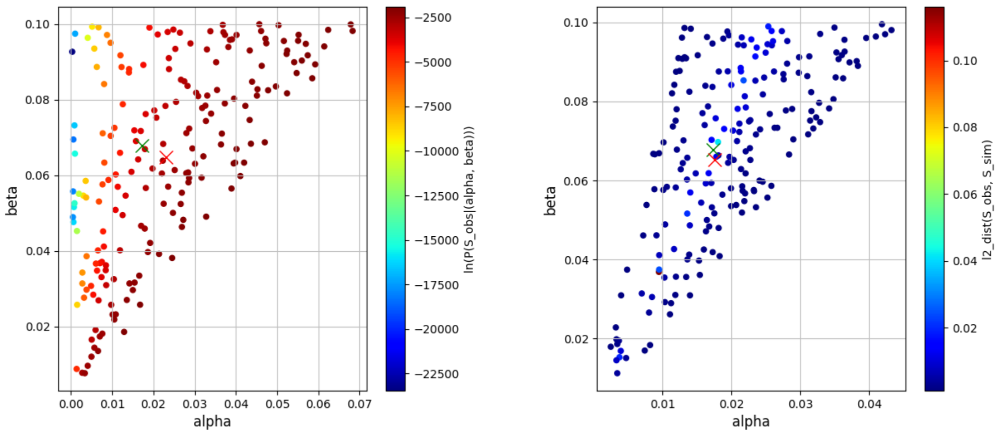


(a) Sampled particles using Rational Functions SMC

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

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

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

[11] 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.

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

[13] 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.

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

[15] 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.

[16] 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.

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

[18] 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).

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