

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

1	Introduction	1
1.1	Motivation	1
1.2	Contribution	1
1.3	Structure of the thesis	2
2	Probabilistic model checking	3
2.1	Markov chain	3
2.1.1	Discrete Time Markov chain	3
2.1.2	Continuous-time Markov chain	4
2.2	Probabilistic temporal logic	5
2.3	Parametric model	5
2.3.1	Parametric Discrete Time Markov chain	5
3	Bayesian inference	7
3.1	Bayesian inference	7
3.1.1	Bayesian formula	7
3.1.2	Bayesian parameter estimation	8
3.1.3	Selection of prior distribution	9
3.1.4	Estimation of posterior distribution	9
3.1.5	Markov chain Monte-Carlo	11
3.1.6	Sequential Monte-Carlo	13
3.1.7	Approximate Bayesian computation	13
4	Related works	14
5	Framework for parameter synthesis.	15
5.1	Model construction	15
5.2	Framework	15

6	Case study	16
6.1	Zeroconf	16
6.1.1	System description	16
6.1.2	Parametric model	16
6.1.3	Properties	16
6.1.4	Parameter synthesis	16
6.2	Defense bees	16
6.2.1	System description	16
6.2.2	Parametric model	16
6.2.3	Parameter synthesis	16
7	Conclusion	17
7.1	Summary	17
7.2	Future works	17

Acknowledgements

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

Abstract

something

Chapter 1

Introduction

1.1 Motivation

Markov population model is widely used in modeling several systems, for example

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

A limit of Markov population models, such as Discrete-time Markov chain, is that all transition probabilities are known a-priori. In order to formalize unknown attributes of a system, we introduce parameter, hence we have parametric Markov population models. In this thesis, we work with parametric discrete-time Markov chain, specifically data-driven parameter synthesis for problem on 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 [11]. The question this thesis answers is that, given a parametric discrete-time

1.2 Contribution

This thesis surveys the parameter synthesis of discrete time markov population model towards a certain property. We model a system discrete time model of a system.

1.3 Structure of the thesis

- **Chapter 1** introduces motivations and background 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** reviews the state-of-the-art works of other researchers on the problem of parameter synthesis.
- **Chapter 4** describes the method.
- **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 Discrete time Markov chain
- Parameter synthesis problem

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-empty set of *states*

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

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

- $s_{init} : S \rightarrow [0, 1]$ 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.

2.1.2 Continuous-time Markov chain

Continuous-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 Continuous-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-empty set of *states*
- $\mathbf{P} : S \times S \rightarrow [0, 1]$ is the *transition probability* function such that

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

- $\mathbf{r} : S \rightarrow \mathbb{N}$ 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 Parametric model

We introduce parameters to formalize unknown attributes of the system.

Definition 2.3.1 (Polynomial ring). Given a tuple $\mathbf{x} = (x_1, \dots, x_n)$ be a tuple

Definition 2.3.2. 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} .

2.3.1 Parametric Discrete Time Markov chain

With the set of rational functions formally defined, we define parametric Discrete-time Markov chain based the definition on [10].

Definition 2.3.3 (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-empty 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 $(1-\alpha)$.

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

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 [14]. 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

- 1: **procedure** COMPUTE HPD(S)
 - 2: Compute interval width $w = |S| * \alpha$
 - 3: Find modal (peak) of sample points.
 - 4: Return minimal interval of size $|S| - w$ which contains the modal.
 - 5: **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 lead to a false propagation of beliefs and degrade inference results. It is suggested by [13] 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

Posterior conjugation

Conjugated posteriors are special cases of Bayesian inference, in which the prior and posterior distribution belongs to the same family of distribution. Conjugated posteriors give us significant benefits

1. Tractability: we have analytical form of posterior distribution.
2. Computationally effective: updating model parameter is of linear time to the dimension of parameter.

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 $Beta(\alpha + \sum_{i=1}^n x_i, \beta + nk - \sum_{i=1}^n x_i)$ \square

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

$$P(x_1, \dots, x_n | N, \theta_1, \dots, \theta_n) = \frac{n!}{x_1! \dots x_n!} \prod_{i=1}^n \theta_i^{x_i}$$

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 $Dirichlet(\alpha_1 + x_1, \dots, \alpha_n + x_n)$ \square

More detailed description in these cases can be found in [16] and [4]. We summarize the necessary results in the following table:

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

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.

```
1: procedure METROPOLIS-HASTINGS( $D$ , maxIteration)
2:   Select a proposal distribution  $\pi(\theta)$ 
3:   Draw a random initial point  $\theta$ 
4:   Init empty trace  $Trace$ 
5:   while maxIteration not reached do
6:      $L \leftarrow P(D|\theta)$ 
7:     Draw a point  $\theta'$  from the proposal distribution.
8:      $L' \leftarrow P(D|\theta')$ 
9:     if  $\ln(L') - \ln(L) > 0$  then
10:      Add  $\theta'$  to  $Trace$ 
11:       $\theta = \theta'$ 
12:     else
13:       Draw a random number  $x$  from  $Uniform(0, 1)$ 
14:       if  $x \leq \xi$ , ( $\xi$  very small, e.g  $10^{-8}$ ) then
15:        Add  $\theta'$  to  $Trace$  (avoiding local maxima)
16:         $\theta = \theta'$ 
17:       end if
18:     end if
19:   end while
20: 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 effi-

cient 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

here [5] [15]

3.1.7 Approximate Bayesian computation

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. [11] briefly summarized important aspect of probabilistic model checking.

Polgreen et al [13] presents a method for bayesian inference of pMC parameters in

The definition and model checking of DTMC and pMC is studied by [2], [8], and [11].

Bayesian inference of pMC parameters is studied in [13] and [9]. In [13], the authors developed methods to synthesize parameters to satisfy a given set of PCTL properties. In [9], the authors presented methods to perform model checking of biological system using Bayesian statistic. The authors in [9] 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 [7].

In this project, we use bee colony model semantics from [6]. The methods and implementation in this project is designed to extend the results of [6] and its tool *DiPS*

storm drawback: it does not support discrete event simulation

In [12] the author introduces the same approach but it is to use on CSL properties and CTMC.

Chapter 5

Framework for parameter synthesis.

5.1 Model construction

5.2 Framework

Chapter 6

Case study

6.1 Zeroconf

6.1.1 System description

Zero configuration protocol is as protocol widely used in the internet

6.1.2 Parametric model

6.1.3 Properties

6.1.4 Parameter synthesis

6.2 Defense bees

6.2.1 System description

6.2.2 Parametric model

6.2.3 Parameter synthesis

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] Matej Hajnal et al. “Data-Informed Parameter Synthesis for Population Markov Chains”. In: *International Workshop on Hybrid Systems Biology*. Springer. 2019, pp. 147–164.
- [7] Faraz Hussain et al. “Automated parameter estimation for biological models using Bayesian statistical model checking”. In: *BMC bioinformatics* 16.S17 (2015), S8.
- [8] Lisa Hutschenreiter, Christel Baier, and Joachim Klein. “Parametric Markov chains: PCTL complexity and fraction-free Gaussian elimination”. In: *arXiv preprint arXiv:1709.02093* (2017).
- [9] 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.
- [10] Sebastian Junges et al. “Parameter synthesis for Markov models”. In: *arXiv preprint arXiv:1903.07993* (2019).

- [11] 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.
- [12] 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.
- [13] 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.
- [14] John Salvatier, Thomas V Wiecki, and Christopher Fonnesbeck. “PyMC3: Python probabilistic programming framework”. In: *ascl* (2016), ascl–1610.
- [15] 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).
- [16] Stephen Tu. “The dirichlet-multinomial and dirichlet-categorical models for bayesian inference”. In: *Computer Science Division, UC Berkeley* (2014).