Bayesian parameter synthesis for Markov population models.

Phung Nhat-Huy

University of Konstanz

May 17, 2021

Universität Konstanz

Motivation.

Motivation

- We study the population dynamics of a system of interest. For example:
  - Number of online nodes in a computer network.
  - Number of surviving individuals in an epidemic model.
- We study the population in a grey-box setup
  - Estimating the model's unknown attributes with experiment data of the population at its steady state.

Motivation

- Modeling population using a stochastic process (Markov population model [12])
  - Discrete-time Markov chain
- Parameterization: encoding the unknown attributes of a system by model parameters
  - Parametric discrete-time Markov chain [10]).

Motivation

As parameters represent unknown features of the system, it gives the following research questions

- *(Parameter inference):* Given a set of data collected by observing the system, what can we know about its parameters?
- *(Parameter synthesis):* Which values of parameters instantiate a model that satisfies a specific property of interest?

Probabilistic model checking

Discrete-time Markov chain

Definition (Discrete Time Markov Chain [1])

A *Discrete-time Markov chain* (or DTMC in short) $\mathcal{M}$ is a tuple $(S, \mathbf{P}, \iota_{init}, AP, L)$, in which

- S is a countable, non-emty set of *states*
- $\mathbf{P} : S \times S \to [0, 1]$ is the *transition probability* function such that
$$\forall s \in S : \sum_{s' \in S} \mathbf{P}(s, s') = 1$$
- $\iota_{init} : S \to [0, 1]$ is the *initial distribution* such that
$$\sum_{s \in S} \iota_{init}(s) = 1$$
- AP is a set of *atomic propositions*.
- $L : S \to 2^{AP}$ is the labelling function on states.

Bottom Strongly Connected Components

Definition (Strongly Connected Component)

Let $\mathcal{M} = (S, \mathbf{P}, \iota_{init}, AP, L)$ be 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 states in S'. If S' has no superset $S'' \subseteq S$, such that S'' is strongly connected, then S' is a *Strongly Connected Component*, or *SCC* in short.

Definition (Bottom Strongly Connected Component)

Let $\mathcal{M} = (S, \mathbf{P}, \iota_{init}, AP, L)$ be a DTMC and $S' \in S$ a Strongly Connected Component. S' is also a *Bottom Strongly Connected Component* (or *BSCC* in short), if and only if there exist no state $s \in S \backslash S'$ that is reachable from any state in S'. If $|S'| = 1$ then S' is a *trivial BSCC*. We denote $BSCC(\mathcal{M}) \in S$ is the set of all BSCCs of $\mathcal{M}$.

Example of DTMC

Algorithm by Knuth and Yao [13] to model a fair dice by a fair coin.

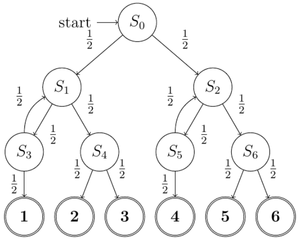


Figure: DTMC model of Knuth-Yao die. There are six BSCCs labbeled "1" to "6"

Probabilistic Computational Tree Logic

Definition (PCTL [1])

The syntax of PCTL consists of state formulas and path formulas.

- State formulas are defined over AP

$$\Phi ::= \text{true} \mid a \mid \Phi \mid \Phi_1 \wedge \Phi_2 \mid \Phi_1 \vee \Phi_2 \mid P_J(\phi)$$

  where $a \in AP$, ϕ is a path formula, and $J \subseteq [0, 1]$ is an interval.

- Path formulas

$$\phi ::= \bigcirc\Phi \mid \Phi_1 \mathsf{U} \Phi_2 \mid \Phi_1 \mathsf{U}^{\leq n} \Phi_2$$

  where Φ, Φ_1, Φ_2 are state formulas, and $n \in \mathbb{N}$.

Example of DTMC

Algorithm by Knuth and Yao [13] to model a fair dice by a fair coin.

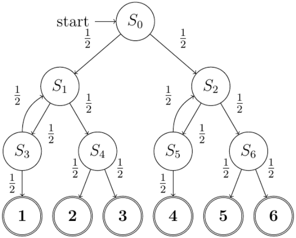


Figure: DTMC model of Knuth-Yao die. There are six BSCCs labbeled "1" to "6"

The probability that the simulation eventually ends with the outcome "one dot" is equal to $\frac{1}{6}$:

$$P_{=\frac{1}{6}}(\mathtt{True U "1"})$$

Parametric discrete-time Markov chain

Definition (Parametric discrete-time Markov chain [8])

A *parametric discrete-time Markov chain* $\mathcal{M}_\theta$ is a tuple $(S, \theta, \mathbf{P}, \iota_{init}, AP, L)$ where

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

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

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

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

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

Example of DTMC

Algorithm by Knuth and Yao [13] to model a possibly unfair dice by two possibly unfair coins.

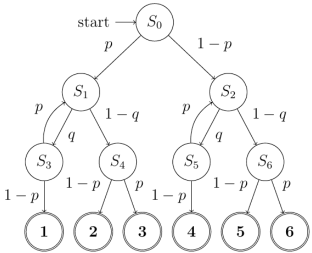


Figure: DTMC model of Knuth-Yao die with two possibly unfair coins. There are six BSCCs labbeled "1" to "6"

Parameter synthesis

Definition (Parameter synthesis (Katoen [10]))

Given a finite-state parametric Markov model, find the parameter values for which a given reachability property exceeds (or is below) a given threshold β.

Parameter synthesis

Katoen [9] summarizes the following methods on parameter synthesis of parametric DTMC:

1. *Computing symbolic reachability probabilities (Daws [4], Hahn [7])*: symbolic solving system of linear equations.
2. *Candidate region generation and checking (Kwiatkowska [14])*: partition the parameter space into *safe* and *unsafe* regions.
3. *Parameter lifting ([15])* replace parametric transition system by a non-parametric one with transition labels are bounds from given intervals.

Example of DTMC

Given DTMC model $\mathcal{M}_{(p,q)}$ as in 3 and a property

$$\Phi = P_{\geq 0.2}(\texttt{TrueU"1"})$$

A Monte Carlo sampling using $p, q \sim \mathit{Uniform}(0, 1)$ gives the following satisfying parameter values

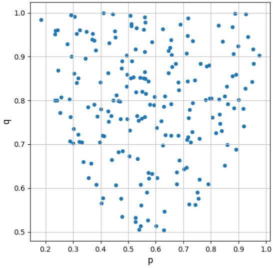


Figure: Samples of (p, q) that instantiate $\mathcal{M}_{(p,q)} \models \Phi$.

Bayesian inference

Bayes theorem

Definition (Bayes theorem)

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

where

- $\pi(\theta)$ is the *prior distribution*.
- $P(D_{obs}|\theta)$ is the *likelihood*.
- $\int_{\theta} P(D_{obs}|\theta)\pi(\theta)d\theta$ is the *marginal distribution*.
- $\pi(\theta|D_{obs})$ is the *posterior distribution*

Bayesian parameter estimation

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

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

- In case we have samples from posterior distribution, for example a set of N parameter values $(\theta_1, \ldots, \theta_N)$ sampled from the posterior distribution $\pi(\theta|D_{obs})$, the discrete form of posterior mean is used:

$$\hat{\theta} \approx \mathbf{E}[\theta] \approx \sum_{\theta} \theta\pi(\theta|D_{obs})$$

Approximation of posterior distribution

- Usually posterior distribution $\pi(\theta|D_{obs})$ has no analytical to evaluate
- We use sampling algorithms to draw samples from posterior distribution
  - Metropolis-Hastings
  - Sequential Monte-Carlo
  - Approximate Bayesian computation

Metropolis-Hastings

Algorithm Metropolis-Hastings Algorithm

procedure METROPOLIS-HASTINGS
    Init empty trace
    Init θ_1 from $\pi(\theta)$
    while $i \leq N_{MH}$ **do**
        Draw θ_{cand} from $Q(\theta'|\theta_{i-1})$
        Compute $\xi = \min(0, \ln(P(D_{obs}|\theta_{cand})) - \ln(P(D_{obs}|\theta_{i-1})))$
        if $\xi > 0$ **then**
            Accept θ_{cand}, append to trace.
        else
            Draw a random number u from $Uniform(0, 1)$
            if $u \leq \exp(\xi)$ **then**
                Accept θ_{cand}, append to trace.
    Return $(\theta_1, \ldots, \theta_N)$, $(w_1, \ldots, w_N)$

Metropolis-Hastings

Advantages of Metropolis-Hastings are:

+ Parameter transition only needs the computation of the likelihood function.
+ Computationally efficient, as marginal distribution is canceled out, and likelihood can be replaced by log-likelihood.
+ Simple to implement.

Disadvantages of Metropolis-Hastings are

- Highly probable to be stuck in a local maximum or minimum.
- Not parallelizable

Sequential Monte-Carlo

Algorithm Sequential Monte Carlo Algorithm

procedure SEQUENTIAL-MONTE CARLO
  Draw $(\theta_1, \ldots, \theta_N)$ from $\pi(\theta)$ ▷ SMC initialization
  $t \leftarrow 1$
  while $t \leq M$ **do**
    Normalize $(w_1^{t-1}, \ldots, w_N^{t-1})$ ▷ SMC correction step
    Sample with replacement $(\theta_1^t, \ldots, \theta_N^t)$ ▷ SMC selection step
      from $(\theta_1^{t-1}, \ldots, \theta_N^{t-1})$ with probabilities $(w_1^{t-1}, \ldots, w_N^{t-1})$
    $i \leftarrow 1$
    while $i \leq N$ **do** ▷ SMC perturbation step
      Draw $\hat{\theta}_i^t$ from $F_t(\theta^t | \theta_1^{t-1}, \ldots, \theta_N^{t-1}), 1 \leq t \leq M$
      Mutate $(\theta_1^*, \ldots, \theta_{N_{MH}}^*), (w_1^*, \ldots, w_{N_{MH}}^*) \leftarrow MH(\hat{\theta}_i^t)$
      $\theta_i \leftarrow \theta_{N_{MH}}^*$
      $w_i \leftarrow w_{N_{MH}}^*$
  Return $(\theta_1, \ldots, \theta_N)$, $(w_1, \ldots, w_N)$

Sequential Monte-Carlo

Sequential Monte Carlo algorithm has several advantages compared to Metropolis-Hastings algorithm.

+ Approximate multimodal distributions: N particles moving independently.
+ Parallelizable.

However, Sequential Monte Carlo also has disadvantages:

- Selection of perturbation and transition kernel is not trivial (Filippi [6], and Silk [16]).
- More difficult to implement.

Approximate Bayesian Computation

A likelihood-free method

- used when likelihood has no analytical form, or the analytical form is expensive to be evaluated.
- estimates the likelihood $P(D_{obs}|\theta)$, or replace it by other measures.

Approximate Bayesian Computation

Algorithm Approximate Bayesian Computation

procedure APPROXIMATE-BAYESIAN-COMPUTATION
    Select a proposal distribution $\pi(\theta)$
    $i \leftarrow 1$
    while $i \leq N$ **do**
        Draw a random particle θ from $\pi(\theta)$
        Simulate data D_{sim} from $\mathcal{M}_\theta$
        if $d = \delta(D_{sim}, D_{obs}) < \epsilon$ **then**
            $\theta_i \leftarrow \theta$
            $w_i = d$
    Return $(\theta_1, \ldots, \theta_N)$, $(w_1, \ldots, w_N)$

Approximate Bayesian Computation

Advantages of Approximate Bayesian Computation are:

+ Likelihood-free: applicable when the likelihood has no analytical form or there is no likelihood.
+ Easy to implement.

However, Approximate Bayesian Computation has drawbacks:

- How to select a distance threshold ϵ so that the posterior is closely approximated? [17]
- How to choose a summary statistic to capture sufficient information? [3]

Parameter synthesis and Parameter inference

What are the differences between parameter inference and parameter synthesis?

	Input	Ouput
Parameter inference	Model $\mathcal{M}_\theta$ Observed data D_{obs}	Parameter estimation $\hat{\theta}$
Parameter synthesis	Model $\mathcal{M}_\theta$ Reachability property Φ	$(\theta_1, \dots, \theta_N)$ $\forall \theta_i \in (\theta_1, \dots, \theta_N)$: $\mathcal{M}_{\theta_i} \models \Phi$

Contributions

Framework design

- In this thesis, we combine parameter synthesis and parameter inference into a data-informed parameter synthesis framework.
  - Sample a set of parameter values which satisfy a property of interest *(parameter synthesis)* and use it to estimate a parameter value which the observed data is likely to be simulate from *(parameter inference)*.

Framework design

- Given experiment data of a system at its steady state and a property of interest
  - *Bayesian parameter inference:* apply different sampling algorithm to approximate the posterior distribution of parameter.
  - *Parameter synthesis:* only accept the sampled points which satisfy the property of interest.
  - *Paramter estimation:* from sampled point, compute an estimation of model parameter.

Challenges

State explosion

The *state-explosion problem* occurs when the size of a model state space grows exponentially as the number of state variables in the system increases [2].

The state-space explosion problem renders probabilistic model checking computationally expensive. We cope this problem using 2 different strategies:

- Pre-compute symbolic reachability probability *(rational function)*.
- Estimate reachability probability statistically *(statistical model checking)*.

Challenges

Approximation of posterior distribution

In Bayesian parameter inference, the posterior distribution usually does not have an analytical form. Hence, we approximate the posterior distribution using different sampling algorithms.

Challenges are:

- How to select a sampling algorithm?
- How to select parameters for the sampling algorithm?

Contribution

- Designed and implemented a data-informed, Bayesian framework on parameter synthesis of parametric Discrete-time Markov chain.
    - when the exact likelihood function of the property of interest is available *RF-SMC*, and
    - when it has to be approximated by simulations *SMC-ABC-SMC*.

Contribution

- Designed and implemented a data-informed, Bayesian framework on parameter synthesis of parametric Discrete-time Markov chain.
  - when the exact likelihood function of the property of interest is available *RF-SMC*, and
  - when it has to be approximated by simulations *SMC-ABC-SMC*.
- Compared the performances (accuracy and scalability) of proposed frameworks on different case studies and different model state-space sizes.

Frameworks

Frameworks

- Generic framework
- RF-SMC
- SMC-ABC-SMC

Generic framework

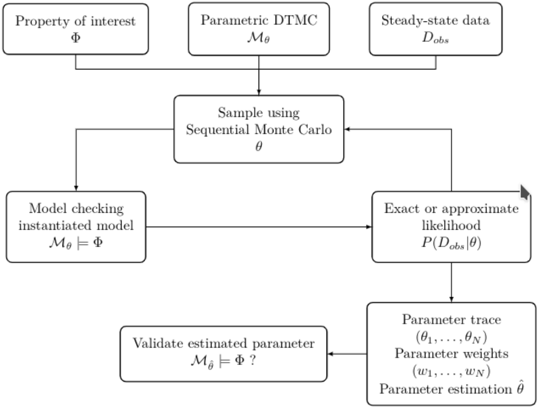


Figure: Generic framework for Bayesian parameter synthesis of parametric DTMC.

Generic framework

- Input:
  - $\mathcal{M}_\theta$: parametric DTMC of parameter θ
  - Φ: bounded reachability property of interest.
  - D_{obs}: observed data.
  - N: number of particles.
- Output:
  - $(\theta_1, \ldots, \theta_{N_{MH}})$, sampled particles
  - $(w_1, \ldots, w_{N_{MH}})$ particle corresponding weights.
  - $\hat{p} = P(\mathcal{M}_{\hat{\theta}} \models \Phi)$

Generic framework

Algorithm Generic framework for Bayesian parameter synthesis

```
procedure GENERIC-BAYESIAN-MONTE-CARLO
    i ← 1
    while i ≤ N do
        Sample θ with corresponding weight w
                by Sequential Monte Carlo sampling algorithm.
        Verify instantiated model M_θ against Φ
        if M_θ ⊨ Φ then
            θ_i ← θ
            Estimate w_i as exact or approximated likelihood P(D_obs|θ)
    Estimate θ̂ using posterior mean.
    Compute p̂ = P(M_θ̂ ⊨ Φ)
    Return (θ_1, ..., θ_N), (w_1, ..., w_N), θ̂, p̂
```

RF-SMC

- Based on Sequential Monte-Carlo
- Model checking using precomputed symbolic reachability probability at each SMC perturbation step.
  - At each MH transition, a particle is accepted only if it instantiates a satisfying model.

RF-SMC

Algorithm Metropolis-Hastings with rational functions

procedure RF-MH
- Init empty trace T
- Draw θ_{cand} from $\pi(\theta)$ s.t $\mathcal{M}_{\theta_{cand}} \models \Phi$ (evaluating $RF_\Phi(\theta)$)
- Append θ_{cand} to trace T
- $i \leftarrow 2$
- **while** $i \leq N_{MH}$ **do**
  - Draw θ_{cand} from $Q(\theta'|\theta_{i-1})$ s.t $\mathcal{M}_{\theta_{cand}} \models \Phi$ (evaluating $RF_\Phi(\theta)$)
  - Evaluate $\xi = \min(0, \ln(P(D_{obs}|\theta_{cand})) - \ln(P(D_{obs}|\theta_{i-1}))) > 0$
  - **if** $\xi > 0$ **then**
    - Append θ_{cand} to trace T
  - **else**
    - Draw a random number u from $Uniform(0, 1)$
    - **if** $u \leq \exp(\xi)$ **then**
      - Append θ_{cand} to trace T
- Return $(\theta_1, \ldots, \theta_{N_{MH}})$, $(w_1, \ldots, w_{N_{MH}})$

RF-SMC

Algorithm Sequential Monte Carlo with rational functions

procedure RF-SMC
- Draw $(\theta_1, \ldots, \theta_N)$ from $\pi(\theta)$ s.t $\mathcal{M}_{\theta_i} \models \Phi$ (evaluating $RF_\Phi(\theta)$)
- $t \leftarrow 1$
- **while** $t \leq M$ **do**
  - Normalize $(w_1^{t-1}, \ldots, w_N^{t-1})$ ▷ SMC correction step
  - Sample with replacement $(\theta_1^t, \ldots, \theta_N^t)$ ▷ SMC selection step
    from $(\theta_1^{t-1}, \ldots, \theta_N^{t-1})$ with probabilities $(w_1^{t-1}, \ldots, w_N^{t-1})$
  - $i \leftarrow 1$
  - **while** $i \leq N$ **do** ▷ SMC perturbation step
    - Draw $\hat{\theta}_i^t$ from $F_t(\theta^t | \theta_1^{t-1}, \ldots, \theta_N^{t-1}), 1 \leq t \leq M$
    - Mutate $(\theta_1^*, \ldots, \theta_{N_{MH}}^*), (w_1^*, \ldots, w_{N_{MH}}^*) \leftarrow RF - MH(\hat{\theta}_i^t)$
    - $\theta_i, w_i \leftarrow \theta_{N_{MH}}^*, w_{N_{MH}}^*$
- Estimate $\hat{\theta}$ using posterior mean, compute $\hat{p} = P(\mathcal{M}_{\hat{\theta}} \models \Phi)$
- Return $(\theta_1, \ldots, \theta_N)$, $(w_1, \ldots, w_N)$, $\hat{\theta}$, $\hat{p}$

SMC-ABC-SMC

Without the availability of analytical form to evaluate the steady-state distribution and the property of interest, we face the following obstacles:

- **Absence of likelihood functions**
  - no analytical form of likelihood.
  - $\Rightarrow$ likelihood-free methods (Approximate Bayesian Computation)
- **Absence of rational functions for evaluation of property of interest**
  - $\Rightarrow$ Statistical model checking

SMC-ABC-SMC

Algorithm Sequential Monte-Carlo with simulations

procedure SMC-ABC-SMC
  Draw $(\theta_1, \ldots, \theta_N)$ from $\pi(\theta)$ s.t $\mathcal{M}_{\theta_i} \models \Phi$ (Statistical MC)
  $t \leftarrow 1$
  while $t \leq M$ **do**
    Normalize $(w_1^{t-1}, \ldots, w_N^{t-1})$ ▷ SMC correction step
    Sample with replacement $(\theta_1^t, \ldots, \theta_N^t)$ ▷ SMC selection step
      from $(\theta_1^{t-1}, \ldots, \theta_N^{t-1})$ with probabilities $(w_1^{t-1}, \ldots, w_N^{t-1})$
    $i \leftarrow 1$
    while $i \leq N$ **do** ▷ SMC perturbation step
      Draw $\hat{\theta}_i^t$ from $F_t(\theta^t | \theta_1^{t-1}, \ldots, \theta_N^{t-1}), 1 \leq t \leq M$
      if Statistical Model Checking $\mathcal{M}_{\hat{\theta}_i^t} \models \Phi$ **then**
        Simulate D_{sim} from $\mathcal{M}_{\hat{\theta}_i^t}$
        if $\delta = Distance(D_{sim}, D_{obs}) < \epsilon)$ **then**
          $\theta_i, w_i \leftarrow \hat{\theta}_i^t, \delta^{-1}$
  Estimate $\hat{\theta}$ using posterior mean, compute $\hat{p} = P(\mathcal{M}_{\hat{\theta}} \models \Phi)$
  Return $(\theta_1, \ldots, \theta_N)$, $(w_1, \ldots, w_N)$, $\hat{theta}$, $\hat{p}$

Comparison

Algorithm 7 Sequential Monte-Carlo with simulations

```
procedure SMC-ABC-SMC
    Draw (θ_1, ..., θ_N) from π(θ) s.t M_{θ_i} ⊨ Φ (Statistical MC)
    t ← 1
    while t ≤ M do
        Normalize (w_1^{t−1}, ..., w_N^{t−1})                    ▷ SMC correction step
        Sample with replacement (θ_1^t, ..., θ_N^t)             ▷ SMC selection step
            from (θ_1^{t−1}, ..., θ_N^{t−1}) with probabilities (w_1^{t−1}, ..., w_N^{t−1})
        i ← 1
        while i ≤ N do                                          ▷ SMC perturbation step
            Draw θ̂_i^t from F_t(θ^t | θ_1^{t−1}, ..., θ_N^{t−1}), 1 ≤ t ≤ M
            if Statistical Model Checking M_{θ̂_i^t} ⊨ Φ then
                Simulate D_sim from M_{θ̂_i^t}
                if δ = Distance(D_sim, D_obs) < ε then
                    θ_i, w_i ← θ̂_i^t, δ^{−1}
    Estimate θ̂ using posterior mean, compute p̂ = P(M_θ̂ ⊨ Φ)
    Return (θ_1, ..., θ_N), (w_1, ..., w_N), thêta, p̂
```

Algorithm 6 Sequential Monte Carlo with rational functions

```
procedure RF-SMC
    Draw (θ_1, ..., θ_N) from π(θ) s.t M_{θ_i} ⊨ Φ (evaluating RF_Φ(θ))
    t ← 1
    while t ≤ M do
        Normalize (w_1^{t−1}, ..., w_N^{t−1})                    ▷ SMC correction step
        Sample with replacement (θ_1^t, ..., θ_N^t)             ▷ SMC selection step
            from (θ_1^{t−1}, ..., θ_N^{t−1}) with probabilities (w_1^{t−1}, ..., w_N^{t−1})
        i ← 1
        while i ≤ N do                                          ▷ SMC perturbation step
            Draw θ̂_i^t from F_t(θ^t | θ_1^{t−1}, ..., θ_N^{t−1}), 1 ≤ t ≤ M
            Mutate (θ_1^*, ..., θ_{N_MH}^*), (w_1^*, ..., w_{N_MH}^*) ← RF − MH(θ̂_i^t)
            θ_i, w_i ← θ_{N_MH}^*, w_{N_MH}^*
    Estimate θ̂ using posterior mean, compute p̂ = P(M_θ̂ ⊨ Φ)
    Return (θ_1, ..., θ_N), (w_1, ..., w_N), θ̂, p̂
```

Figure: Comparison between RF-SMC and SMC-ABC-SMC.

Evaluation

Evaluation

Case studies:

- Zeroconf
- Social feedback in honeybee colonies
- SIR

Evaluation environment:

- Hardware: Intel Xeon W-2135, 64GB RAM
- OS: OpenSUSE 15.2
- Libraries: Storm @stable, PRISM 4.6, Python 3.8.8

Zeroconf

Zero-configuration protocol (*zeroconf* for short) [5] is a protocol used in IPv4 network to allocate newly attached device an unique IP address without any intervention from network operators.

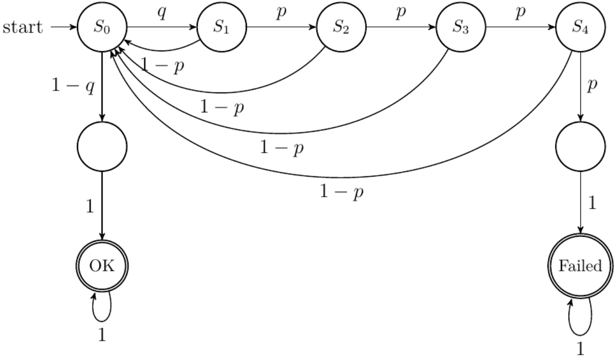


Figure: Example of an IPv4 Zeroconf model with 4 probes

Zeroconf

We select a true parameter (p, q) arbitrarily random for Zeroconf model of 4 and 10 states.

Model $\mathcal{M}$	Zeroconf, 4 probes	Zeroconf, 10 probes
Number of BSCCs	2	2
Number of states	9	14
True parameter $\theta = (p, q)$	(0.105547, 0.449658)	(0.197779, 0.621824)
Number of samples	10000	10000
Synthetic data D_{obs}	(41, 9959)	(22, 9978)
Property of interest Φ	$P_{\geq 0.75}(\texttt{true}\mathrm{U}^{\leq 4}(\texttt{"OK"}))$	$P_{\geq 0.75}(\texttt{true}\mathrm{U}^{\leq 10}(\texttt{"OK"}))$
Satisfaction property $P(\mathcal{M}_\theta \models \Phi)$	0.946409	0.952067

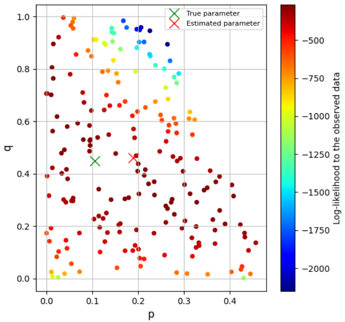
Table: Synthetic data for Zeroconf model of 4 and 10 probes.

Zeroconf 4 results

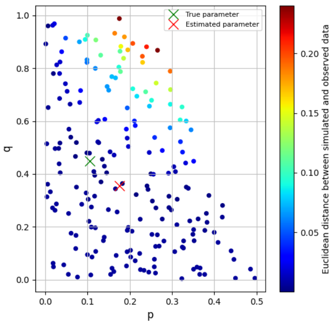
Method	RF-SMC	SMC-ABC-SMC
Estimated parameter $\hat{\theta}$	(0.188956, 0.460554)	(0.176469, 0.355322)
True parameter θ	(0.105547, 0.449658)	(0.105547, 0.449658)
L2 distance $\|\theta, \hat{\theta}\|_2$	0.084117	0.118023
$P(\mathcal{M}_{\hat{\theta}} \models \Phi)$	0.893715	0.918133

Table: Parameter estimation results for Zeroconf model of 4 probes.

Zeroconf 4 results



(a) Sampled particles using Rational Functions SMC



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

Figure: Parameter synthesis results for Zeroconf model of 4 probes.

Zeroconf 4 results

Runtime

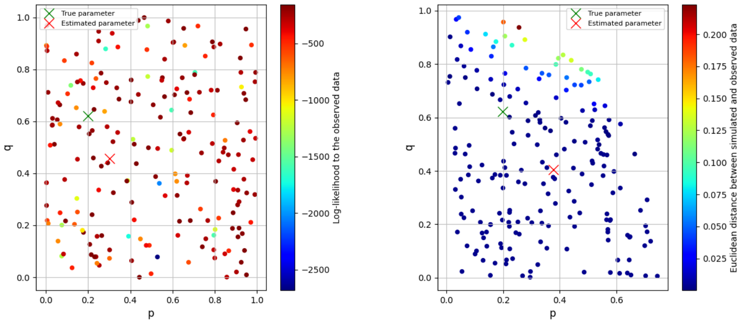
Method	RF-SMC	SMC-ABC-SMC
Total runtime (minutes)	6.083	54.867
Average perturbation runtime (minutes)	0.32	2.88

Table: Physical runtime on Zeroconf model with 4 probes.

Zeroconf 10 results

Method	RF-SMC	SMC-ABC-SMC
True parameter θ	(0.197779, 0.621824)	(0.197779, 0.621824)
Estimated parameter $\hat{\theta}$	(0.301807, 0.457090)	(0.378774, 0.405870)
L2 distance $\|\theta, \hat{\theta}\|_2$	0.194831	0.281772
$P(\mathcal{M}_{\hat{\theta}} \models \Phi)$	0.952067	0.966142

Table: Parameter estimation results for Zeroconf model of 10 probes.



(a) Sampled particles using RF-SMC

(b) Sampled particles using SMC-ABC-SMC

Figure: Parameter synthesis results for Zeroconf model of 10 probes.

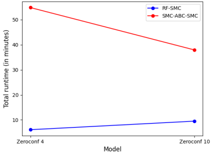
Zeroconf 10 results

Runtime

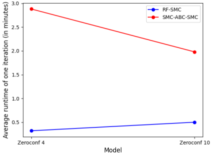
Method	RF-SMC	SMC-ABC-SMC
Total runtime (minutes)	9.50	37.93
Average perturbation runtime (minutes)	0.501	1.978

Table: Physical runtime on Zeroconf model with 10 probes.

Zeroconf results discussion



(a) Total runtime



(b) Average perturbation runtime

Figure: Physical runtime on Zeroconf model of different sizes.

- RF-SMC and SMC-ABC-SMC have similar accuracy.
- RF-SMC is much faster

Social feedback in honeybee colonies

Each bee in a colony possibly stings after observing a threat in the surrounding environment and warns other bees by releasing a special substance, pheromone. There are 3 assumptions on the system:

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

Social feedback in honeybee colonies

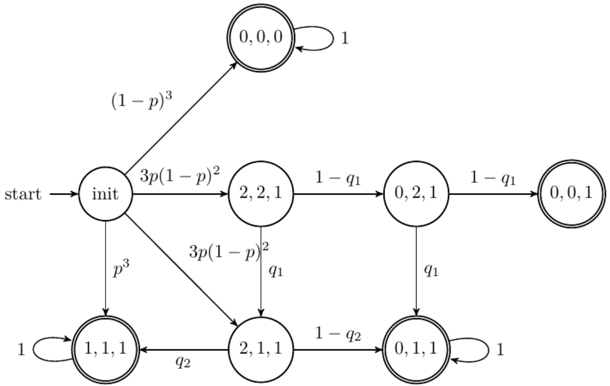


Figure: Parametric DTMC model of 3 bees with 3 parameters p, q_1, q_2

True parameters and synthetic data.

Model $\mathcal{M}$	3 bees	5 bees	10 bees
Number of states	13	24	69
Number of BSCCs	4	6	11
True parameter θ	$p = 0.665623$ $q_1 = 0.830401$ $q_2 = 0.839778$	$p = 0.278370$ $q_1 = 0.305994$ $q_2 = 0.489792$ $q_3 = 0.737252$ $q_4 = 0.766581$	$p = 0.222169$ $q_1 = 0.246993$ $q_2 = 0.281934$ $q_3 = 0.446384$ $q_4 = 0.491612$ $q_5 = 0.534611$ $q_6 = 0.569409$ $q_7 = 0.684651$ $q_8 = 0.717139$ $q_9 = 0.800987$
Synthetic data D_{obs}	(344, 54, 1390, 8212)	(1940, 11, 216, 2682, 4200, 951)	(769, 0, 1, 10, 187, 972, 2494, 2982, 2133, 419, 33)
Target property Φ	$P_{\geq 0.25}(\texttt{trueU}(S > 3))$	$P_{\geq 0.25}(\texttt{trueU}(S > 5))$	$P_{\geq 0.25}(\texttt{trueU}(S > 8))$
$P(\mathcal{M}_\theta \models \Phi)$	0.819666	0.780172	0.737244

Table: True parameter and synthetic data for 3, 5, and 10 bees models.

Social feedback in honeybee colonies

	RF-SMC	SMC-ABC-SMC
True parameter θ	$p = 0.665623$ $q_1 = 0.830401$ $q_2 = 0.839778$	$p = 0.665623$ $q_1 = 0.830401$ $q_2 = 0.839778$
Estimated parameter $\hat{\theta}$	$p = 0.671388$ $q_1 = 0.575026$ $q_2 = 0.525502$	$p = 0.811651$ $q_1 = 0.621073$ $q_2 = 0.544130$
L2 distance $\|\theta, \hat{\theta}\|_2$	0.404992	0.390576
$P(\mathcal{M}_{\hat{\theta}} \models \Phi)$	0.623889	0.595083

Table: Parameter synthesis result for 3 bees model.

Social feedback in honeybee colonies

Method	RF-SMC	SMC-ABC-SMC
Total runtime (minutes)	5.917	68.783
Average perturbation runtime (minutes)	0.312	3.614

Table: Physical runtime on 3 bees model.

Social feedback in honeybee colonies

	RF-SMC	SMC-ABC-SMC
True parameter θ	$p = 0.278370$ $q_1 = 0.305994$ $q_2 = 0.489792$ $q_3 = 0.737252$ $q_4 = 0.766581$	$p = 0.278370$ $q_1 = 0.305994$ $q_2 = 0.489792$ $q_3 = 0.737252$ $q_4 = 0.766581$
Estimated parameter $\hat{\theta}$	$p = 0.576565$ $q_1 = 0.589724$ $q_2 = 0.490334$ $q_3 = 0.554397$ $q_4 = 0.524433$	$p = 0.361220$ $q_1 = 0.316007$ $q_2 = 0.545691$ $q_3 = 0.643962$ $q_4 = 0.591206$
L2 distance $\|\theta, \hat{\theta}\|_2$	0.511366	0.222594
$P(\mathcal{M}_{\hat{\theta}} \models \Phi)$	0.623889	0.595083

Table: Parameter synthesis result for 5 bees model

Social feedback in honeybee colonies

Method	RF-SMC	SMC-ABC-SMC
Total runtime (minutes)	29.517	352.083
Average perturbation runtime (minutes)	1.553	18.518

Table: Physical runtime on 5 bees model.

Social feedback in honeybee colonies

	RF-SMC	SMC-ABC-SMC
True parameter θ	$p = 0.222169$ $q_1 = 0.246993$ $q_2 = 0.281934$ $q_3 = 0.446384$ $q_4 = 0.491612$ $q_5 = 0.534611$ $q_6 = 0.569409$ $q_7 = 0.684651$ $q_8 = 0.717139$ $q_9 = 0.800987$	$p = 0.222169$ $q_1 = 0.246993$ $q_2 = 0.281934$ $q_3 = 0.446384$ $q_4 = 0.491612$ $q_5 = 0.534611$ $q_6 = 0.569409$ $q_7 = 0.684651$ $q_8 = 0.717139$ $q_9 = 0.800987$
Estimated parameter $\hat{\theta}$	$p = 0.604881$ $q_1 = 0.472557$ $q_2 = 0.281484$ $q_3 = 0.500706$ $q_4 = 0.49340$ $q_5 = 0.495508$ $q_6 = 0.466596$ $q_7 = 0.510167$ $q_8 = 0.474153$ $q_9 = 0.484061$	$p = 0.391313$ $q_1 = 0.485688$ $q_2 = 0.424056$ $q_3 = 0.381489$ $q_4 = 0.440681$ $q_5 = 0.578865$ $q_6 = 0.594232$ $q_7 = 0.564557$ $q_8 = 0.547804$ $q_9 = 0.520006$
L2 distance $\|\theta, \hat{\theta}\|_2$	0.665837	0.487042
$P(\mathcal{M}_{\hat{\theta}} \models \Phi)$	0.933287	0.907478

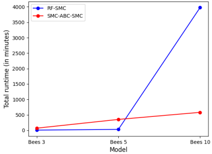
Table: Parameter synthesis result for 10 bees model

Social feedback in honeybee colonies

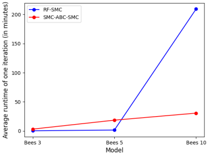
Method	RF-SMC	SMC-ABC-SMC
Total runtime (minutes)	3976.117	581.833
Average perturbation runtime (minutes)	209.237	30.592

Table: Physical runtime on 10 bees model.

Social feedback in honeybee colonies



(a) Total runtime



(b) Average perturbation runtime

Figure: Physical runtime on bees model of different sizes.

Results discussion

- SMC-ABC-SMC delivers results with higher accuracy and comparable satisfaction probability.
- From 10 bees model, RF-SMC becomes much slower than SMC-ABC-SMC.

SIR

SIR model (Kermack [11]) is a model widely used in modeling epidemics. In a SIR model, each individual is of one among three types:

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

SIR is a stochastic system modeled by reactions between S, I and R. In this thesis, we use only two reactions.

$$R_1 : S + I \xrightarrow{\alpha} 2I$$

$$R_2 : I \xrightarrow{\beta} R$$

SIR

Reactions R_1, R_2 generate continuous-time Markov chain (CTMC) depends on the initital population

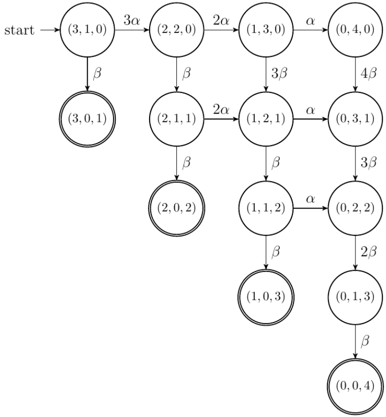


Figure: $SIR(3, 1, 0)$ CTMC model with parameters(α, β).

SIR

We uniformize CTMC into DTMC

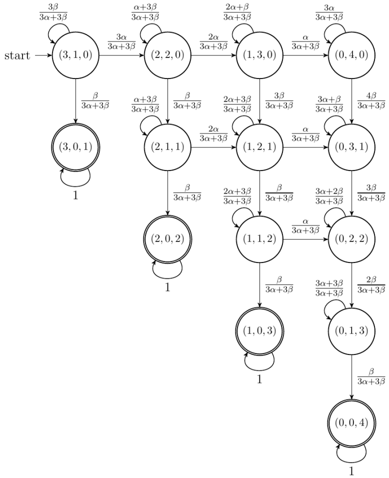


Figure: $SIR(3, 1, 0)$ uniformized DTMC model with uniformization rate $(3\alpha + 4\beta)$

SIR

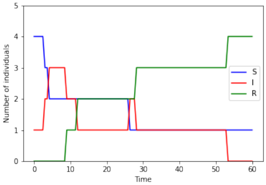
As uniformization of a CTMC preserves bounded until properties, we conduct parameter synthesis experiments on uniformized DTMC. We verify the following property

SIR property of interest

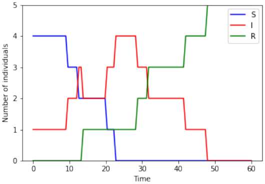
"With the probability of at least 25 percent, the number of infected individuals does not exceed half of the population until the system is in its steady-state. Let $N = S_0 + I_0 + R_0$. In the PCTL formula we have:

$$P_{\geq 0.25}(!(i > N/2) \quad \mathrm{U}^{\leq N} \quad (i = 0))$$

SIR



(a) Example with SIR(5,1,0)



(b) Counter-example with SIR(5,1,0)

Figure: Example and counter-example on SIR(5,1,0) CTMC with $(\alpha, \beta) = (0.034055, 0.087735)$.

SIR

Model $\mathcal{M}$	SIR(5,1,0)	SIR(10,1,0)	SIR(15,1,0)
Number of BSCCs	6	11	16
Number of states	27	77	152
True param. (α, β)	(0.034055, 0.087735)	(0.025490, 0.069298)	(0.011499, 0.062111)
Synthetic data D_{obs}	(1098, 1377, 1296, 1312, 1466, 3451)	(1002, 1258, 1123, 902, 770, 651, 497, 420, 496, 685, 2196)	(50, 181, 302, 455, 539, 567, 582, 566, 541, 553, 574, 528, 512, 586, 875, 2589)
Property Φ	$P_{\geq 0.25}(i \leq 3 \text{ U}^{\leq 6}\ i = 0)$	$P_{\geq 0.25}(i \leq 5 \text{ U}^{\leq 11}\ i = 0)$	$P_{\geq 0.25}(i \leq 8 \text{ U}^{\leq 16}\ i$
$P(\mathcal{M}_{(\alpha,\beta)} \models \Phi)$	0.3474444	0.265815	0.327446

Table: True parameters and synthetic data for SIR(5,1,0), SIR(10,1,0), SIR(15,1,0)

Conclusion

Future work

References I

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

[2] Edmund M Clarke et al. "Model checking and the state explosion problem". In: *LASER Summer School on Software Engineering*. Springer. 2011, pp. 1–30.

[3] Katalin Csilléry et al. "Approximate Bayesian computation (ABC) in practice". In: *Trends in ecology & evolution* 25.7 (2010), pp. 410–418.

[4] Conrado Daws. "Symbolic and parametric model checking of discrete-time Markov chains". In: *International Colloquium on Theoretical Aspects of Computing*. Springer. 2004, pp. 280–294.

References II

[5] *Dynamic Configuration of IPv4 Link-Local Addresses.* Accessed: 2021-03-30. URL: https://tools.ietf.org/html/rfc3927.

[6] 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.

[7] Ernst Moritz Hahn, Holger Hermanns, and Lijun Zhang. "Probabilistic reachability for parametric Markov models". In: *International Journal on Software Tools for Technology Transfer* 13.1 (2011), pp. 3–19.

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

References III

[9] Joost-Pieter Katoen. "Model Checking Meets Probability: A Gentle Introduction.". In: *Engineering dependable software systems* 34 (2013), pp. 177–205.

[10] 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.

[11] William Ogilvy Kermack and Anderson G McKendrick. "A contribution to the mathematical theory of epidemics". In: *Proceedings of the royal society of london. Series A, Containing papers of a mathematical and physical character* 115.772 (1927), pp. 700–721.

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

References IV

[13] Donald E Knuth, KNUTH DE, and YAO AC. "THE COMPLEXITY OF NONUNIFORM RANDOM NUMBER GENERATION.". In: (1976).

[14] Marta Kwiatkowska, Gethin Norman, and David Parker. "Symmetry reduction for probabilistic model checking". In: *International Conference on Computer Aided Verification*. Springer. 2006, pp. 234–248.

[15] Tim Quatmann et al. "Parameter synthesis for Markov models: Faster than ever". In: *International Symposium on Automated Technology for Verification and Analysis*. Springer. 2016, pp. 50–67.

References V

[16] 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).

[17] Scott A Sisson, Yanan Fan, and Mark M Tanaka. "Sequential monte carlo without likelihoods". In: *Proceedings of the National Academy of Sciences* 104.6 (2007), pp. 1760–1765.

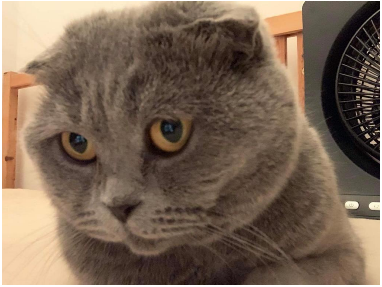


Figure: Thank you for your attention.