

Probabilistic programming for birth-death models of evolution using an alive particle filter with delayed sampling

Jan Kudlicka^a, Lawrence M. Murray^b, Fredrik Ronquist^c, Thomas B. Schön^a

^aUppsala University, ^bUber AI, ^cSwedish Museum of Natural History



UPPSALA
UNIVERSITET



Naturhistoriska
riksmuseet

Uber AI

INTRODUCTION

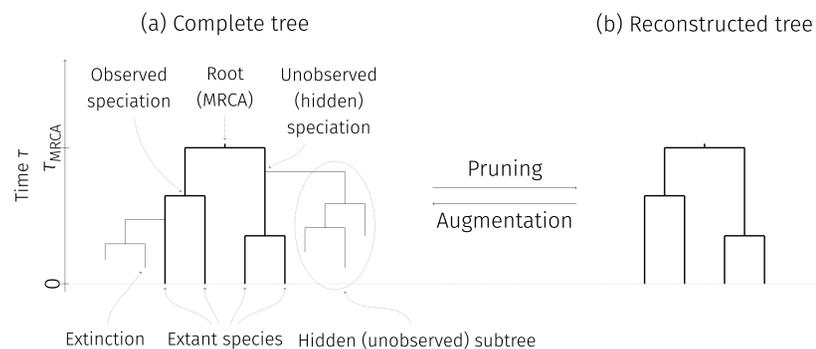
We consider probabilistic programming for birth-death models of evolution and introduce a new widely-applicable inference method that combines an extension of the alive particle filter (APF) with automatic Rao-Blackwellization via delayed sampling. Birth-death models of evolution are an important family of phylogenetic models of the diversification processes that lead to evolutionary trees.

Probabilistic programming languages (PPLs) give phylogeneticists a new and exciting tool: their models can be implemented as probabilistic programs with just a basic knowledge of programming. The general inference methods in PPLs reduce the need for external experts, allow quick prototyping and testing, and accelerate the development and deployment of new models.

BIOLOGY BACKGROUND

Two types of events play a significant role in the evolution:

- *Speciation* occurs when the population of one species splits and eventually forms two new species.
- *Extinction* occurs when the whole population of one species dies out. Species that are not extinct, i.e., species with individuals alive at the present time, are called *extant*.



Birth-death models

- Generative models of evolutionary processes.
- Waiting times between events are exponentially distributed.
- Constant-rate birth-death (CRBD) model – the simplest model with the constant speciation rate λ and the constant extinction rate μ .

BIRTH-DEATH MODELS AS PROBABILISTIC PROGRAMS

We are interested in $P(\theta|T)$, where θ denotes the parameters of the model and T the reconstructed tree (the observation). Birth-death models can be implemented as probabilistic programs that generate proposals by starting with the reconstructed tree and *augmenting* it with unobserved information to obtain a complete tree.

The observed tree is traversed in depth-first order. Along each branch the generative birth-death model is used to simulate

- changes to the state,
- changes to the speciation and extinction rates,
- hidden speciations and hidden subtrees.

We use observe statements to condition on

- no extinction occurring along the branches in the observed tree,
- observed speciations occurring at the end of branches,
- simulated states of the extant species being equal to the observed states,
- no species in the hidden subtrees surviving to the present time.

EXTENDED ALIVE PARTICLE FILTER

Consider using the standard bootstrap particle filter (BPF) for inference. In the simulation of hidden subtrees, if any species survives, the particle is assigned zero weight! In extreme case, all particles have zero weight (the filter degenerates). Del Moral et al. (2015) considered this problem in a setting with indicator potentials: If any particle has zero weight, the resampling and propagation steps are repeated. We have extended the method to work with importance weights.

Standard bootstrap particle filter (BPF)

```

for  $n = 1$  to  $N$  do
   $x_0^{(n)} \leftarrow \emptyset$ ;  $w_0^{(n)} \leftarrow 1/N$ 
end for
for  $t = 1$  to  $T$  do
  for  $n = 1$  to  $N$  do
     $a \sim \text{Cat}(\{w_{t-1}^{(m)} / \sum_{l=1}^N w_{t-1}^{(l)}\}_{m=1}^N)$ 
     $x_t^{(n)} \leftarrow \text{PROPAGATE}(x_{t-1}^{(a)})$ 
     $w_t^{(n)} \leftarrow p(y_t | x_t^{(n)})$ 
  end for
end for

```

Extended alive particle filter (APF)

```

for  $n = 1$  to  $N$  do
   $x_0^{(n)} \leftarrow \emptyset$ ;  $w_0^{(n)} \leftarrow 1/N$ 
end for
for  $t = 1$  to  $T$  do
   $P_t \leftarrow 0$ 
  for  $n = 1$  to  $N + 1$  do
    repeat
       $a \sim \text{Cat}(\{w_{t-1}^{(m)} / \sum_{l=1}^N w_{t-1}^{(l)}\}_{m=1}^N)$ 
       $x_t^{(n)} \leftarrow \text{PROPAGATE}(x_{t-1}^{(a)})$ 
       $P_t \leftarrow P_t + 1$ 
       $w_t^{(n)} \leftarrow p(y_t | x_t^{(n)})$ 
    until  $w_t^{(n)} > 0$ 
  end for
end for

```

Unbiased estimate of the marginal likelihood $p(y_{1:T})$:

$$\hat{Z} = \prod_{t=1}^T \frac{\sum_{n=1}^N w_t^{(n)}}{N} \quad \hat{Z} = \prod_{t=1}^T \frac{\sum_{n=1}^N w_t^{(n)}}{P_t - 1}$$

The alive particle filter uses $N + 1$ particles rather than N in order to estimate the marginal likelihood $p(y_{1:T})$ without bias (proof of unbiasedness in the paper).

DELAYED SAMPLING

In a Bayesian setting, the parameters are associated with a prior distribution. Using the gamma distribution as a prior for the rates of speciation, extinction and state change is mathematically convenient since the gamma distribution is a conjugate prior for both the Poisson and the exponential likelihood.

Consider the following prior:

$$v \sim \text{Gamma}(k, \theta) \text{ with } k \in \mathbb{N}$$

When the program needs to make a draw from a Poisson or an exponential distribution, i.e.

$$n \sim \text{Poisson}(v\Delta) \quad \text{or} \quad \delta \sim \text{Exponential}(v),$$

it can instead make a draw from the marginalized distribution:

$$n \sim \text{NegativeBinomial}\left(k, \frac{1}{1 + \Delta\theta}\right) \text{ resp. } \delta \sim \text{Lomax}\left(\frac{1}{\theta}, k\right),$$

and update the distribution for v to

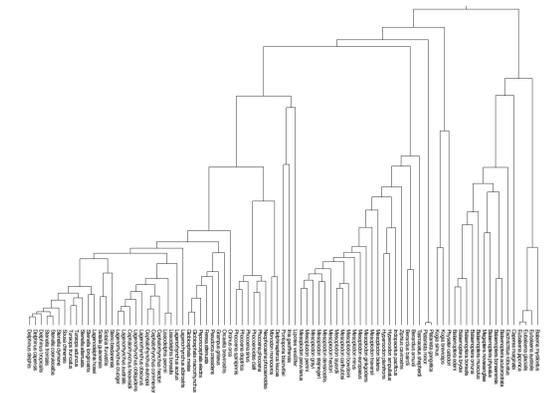
$$v \sim \text{Gamma}\left(k + n, \frac{\theta}{1 + \Delta\theta}\right) \text{ resp. } v \sim \text{Gamma}\left(k + 1, \frac{\theta}{1 + \delta\theta}\right).$$

Using this strategy there is actually *no need to sample the rates at all*. Exploiting the conjugacies can be automated by an algorithm known as *delayed sampling* (Murray et al. 2018).

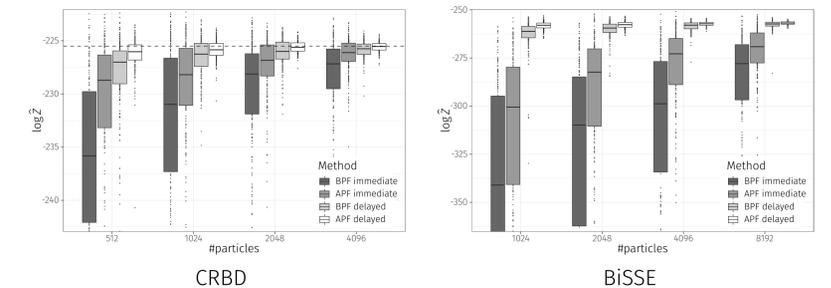
EXPERIMENTS

We implemented the *constant-rate birth-death* (CRBD) model and the *binary state speciation and extinction* (BiSSE) model as programs in Birch (<https://birch-lang.org>). The BiSSE model introduces binary state for species which might change and which is usually observed for the extant species. Waiting times between speciation, extinction, state switch events are exponentially distributed with rates that depend on the state.

We used the phylogeny of cetaceans as the observation. This phylogeny represents the evolution of whales, dolphins and porpoises and contains 87 extant species. For BiSSE model we extended the phylogeny with the state variable related to the body length (small vs. large cetaceans).



We used Gamma(1,1) as the prior for the speciation and extinction rates and Gamma(1, 10/820.28) as the prior for the switch events.



CONCLUSION

We showed how phylogenetic birth-death models can be implemented in probabilistic programming languages and showed the strength of the extended alive particle filter with delayed sampling for these models compared to the standard bootstrap particle filter (BPF). For the BiSSE model using 8192 particles we increased ESS approximately 29 times, CAR approximately 30 times and lowered $\text{var} \log \hat{Z}$ more than 1150 times at the cost of running 3 times more propagations.

The extended APF is a general inference method that shows its strength in situations where particles are often assigned zero weight. In the case when the weights are always positive, the extra cost of using the APF rather than the bootstrap particle filter is negligible, making our method a suitable drop-in replacement for the bootstrap particle filter in probabilistic programming inference.

References

- P. Del Moral, A. Jasra, A. Lee, C. Yau, and X. Zhang. The alive particle filter and its use in particle Markov chain Monte Carlo. *Stochastic Analysis and Applications*, 33(6):943–974, 2015.
- L. M. Murray, D. Lundén, J. Kudlicka, D. Broman, and T. B. Schön. Delayed sampling and automatic Rao-Blackwellization of probabilistic programs. In *International Conference on Artificial Intelligence and Statistics*, pages 1037–1046, 2018.



This research was financially supported by the Swedish Foundation for Strategic Research (SSF) via the project ASSEMBLE and by the Swedish Research Council grants 2013-4853, 2014-05901 and 2017-03807.