Lesson 1: Introduction to Simulation-based Inference for Epidemiological Dynamics

Aaron A. King Edward L. Ionides Translated in pypomp by Kunyang He

2025-07-23

Table of contents

Introduction

Objectives for this lesson

- To understand the motivations for simulation-based inference in the study of epidemiological and ecological systems.
- To introduce the class of partially observed Markov process (POMP) models.
- To introduce the `pomp` `R` package.

What is pypomp

- pypomp is an independent Python framework for building, simulating and fitting partially observed Markov process (POMP) models.

- Written in modern Python 3.10 and built on JAX for just-in-time compilation, automatic differentiation (AD) and transparent CPU/GPU/TPU execution.

- Aims to serve two user groups:

  - Scientists who need fast, plug-and-play likelihood-based inference for nonlinear dynamical systems.

  - Method developers who want a clean, differentiable platform for experimenting with new particle-filter and Bayesian algorithms.

What makes epidemiological inference hard?

Epidemiological and Ecological Dynamics

- Ecological systems are **complex, open, nonlinear, and non-stationary**.
- "Laws of Nature" are unavailable except in the most general form.
- It is useful to model them as **stochastic systems**.
- For any observable phenomenon, **multiple competing explanations** are possible.
- **Central scientific goals**
  - Which explanations are most favored by the data?

  - Which kinds of data are most informative?
- **Central applied goals**
  - How to design ecological or epidemiological intervention?

  - How to make accurate forecasts?
- **Time series** are particularly useful sources of data.

Why pypomp

- **Automatic differentiation**: gradients of log-likelihoods, summary statistics and even full particle-filter traces are available out-of-the-box—no hand-coded adjoints.

- **Hardware acceleration**: a single GPU can advance tens of thousands of particles in parallel, giving up-to-16× speed-ups over CPU-bound workflows.

- **Functional, stateless design**: all simulators take explicit JAX random keys, making large-scale parallelisation and reproducibility straightforward.

Obstacles to inference

*Obstacles for **ecological** modeling and inference via nonlinear mechanistic models enumerated by* (Bjørnstad and Grenfell, 2001):

1. Combining measurement noise and process noise.
2. Including covariates in mechanistically plausible ways.
3. Using continuous-time models.
4. Modeling and estimating interactions in coupled systems.
5. Dealing with unobserved variables.
6. Modeling spatial-temporal dynamics.

The same issues arise for **epidemiological** modeling and inference via nonlinear mechanistic models.

The *partially observed Markov process* (POMP) modeling framework we focus on in this course addresses most of these problems effectively.

Course overview

Course objectives

1. To show how stochastic dynamical systems models can be used as scientific instruments.
2. To teach statistically and computationally efficient approaches for performing scientific inference using POMP models.
3. To give students the ability to formulate models of their own.
4. To give students opportunities to work with such inference methods.
5. To familiarize students with the `pomp` package.
6. To provide documented examples for adaptation and re-use.

Questions and answers

1. How does one combine various data types to quantify asymptomatic COVID-19 infections? (Subramanian et al., 2021)
2. How effective have various non-pharmaceutical interventions been at controlling SARS-CoV-2 spread in hospitals? (Shirreff et al., 2022)
3. How does one use incidence and mobility data to infer key epidemiological parameters? (Andrade and Duggan, 2022)

4. How does one make forecasts for an outbreak of an emerging infectious disease? (King et al., 2015)

5. How does one build a system for real-time surveillance of COVID-19 using epidemiological and mobility data? (Fox et al., 2022)

6. What strategies are effective at containing mumps spread on college campuses? (Shah et al., 2022)

7. What explains the resurgence of pertussis in countries with sustained high vaccine coverage? (Domenech de Cellès et al., 2018)

8. Do subclinical infections of pertussis play an important epidemiological role? (Lavine et al., 2013)

9. Can serotype-specific immunity explain the strain dynamics of human enteroviruses? (Pons-Salort and Grassly, 2018)

10. How does dynamic variation in individual sexual behavior contribute to the HIV epidemic? How does this compare to the role of heterogeneity between individuals? (Romero-Severson et al., 2015)

11. What is the contribution of adults to polio transmission? (Blake et al., 2014)

12. What explains the interannual variability of malaria? (Laneri et al., 2010)

13. Can hydrology explain the seasonality of cholera? (Baracchini et al., 2017)

14. What roles are played by asymptomatic infection and waning immunity in cholera epidemics? (King et al., 2008)

Partially observed Markov processes

Mathematical definitions

Partially observed Markov process (POMP) models

- Data $y_1^*, \dots, y_N^*$ collected at times $t_1 < \dots < t_N$ are modeled as noisy, incomplete, and indirect observations of a Markov process $\{X(t), t \geq t_0\}$.
- This is a *partially observed Markov process (POMP)* model, also known as a hidden Markov model or a state space model.
- $\{X(t)\}$ is Markov if the history of the process, $\{X(s), s \leq t\}$, is uninformative about the future of the process, $\{X(s), s \geq t\}$, given the current value of the process, $X(t)$.
- If all quantities important for the dynamics of the system are placed in the *state*, $X(t)$, then the Markov property holds by construction.

- Systems with delays can usually be rewritten as Markovian systems, at least approximately.
- An important special case: any system of differential equations $dx/dt = f(x)$ is Markovian.
- POMP models can include all the features desired by Bjørnstad and Grenfell (2001).

Schematic of the structure of a POMP

- Arrows in the following diagram show causal relations.
- A key perspective to keep in mind is that *the model is to be viewed as the process that generated the data.*
- That is: the data are viewed as one realization of the model's stochastic process.

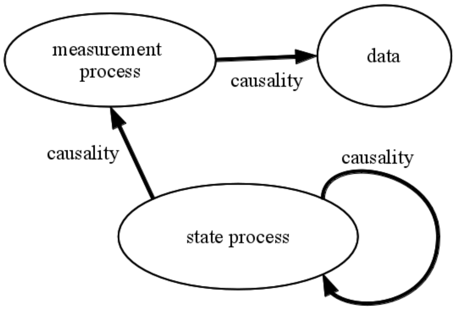


Figure 1: Schematic of the structure of a POMP

Notation for POMP models

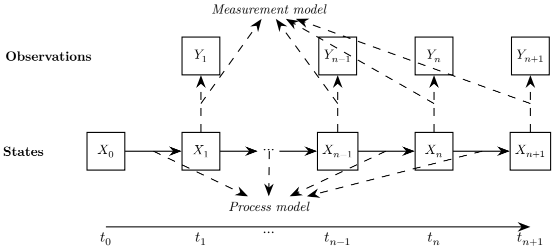
- Write $X_n = X(t_n)$ and $X_{0:N} = (X_0, \ldots, X_N)$. Let Y_n be a random variable modeling the observation at time t_n.
- The one-step transition density, $f_{X_n|X_{n-1}}(x_n|x_{n-1};\theta)$, together with the measurement density, $f_{Y_n|X_n}(y_n|x_n;\theta)$ and the initial density, $f_{X_0}(x_0;\theta)$, specify the entire POMP model.
- The joint density $f_{X_{0:N},Y_{1:N}}(x_{0:N}, y_{1:N};\theta)$ can be written as

$$f_{X_0}(x_0;\theta) \prod_{n=1}^{N} f_{X_n|X_{n-1}}(x_n|x_{n-1};\theta)\, f_{Y_n|X_n}(y_n|x_n;\theta)$$

- The marginal density for $Y_{1:N}$ evaluated at the data, $y^*_{1:N}$, is

$$f_{Y_{1:N}}(y^*_{1:N};\theta) = \int f_{X_{0:N},Y_{1:N}}(x_{0:N}, y^*_{1:N};\theta)\, dx_{0:N}$$

Another POMP model schematic



- The state process, X_n, is Markovian, i.e.,

$$f_{X_n|X_{0:n-1},Y_{1:n-1}}(x_n|x_{0:n-1}, y_{1:n-1}) = f_{X_n|X_{n-1}}(x_n|x_{n-1}).$$

- Moreover, Y_n, depends only on the state at that time:

$$f_{Y_n|X_{0:N},Y_{1:n-1}}(y_n|x_{0:n}, y_{1:n-1}) = f_{Y_n|X_n}(y_n|x_n), \quad \text{for } n = 1, \ldots, N.$$

From math to algorithms

Moving from math to algorithms for POMP models

We specify some *basic model components* which can be used within algorithms:

- `rprocess`: a draw from $f_{X_n|X_{n-1}}(x_n|x_{n-1};\theta)$
- `dprocess`: evaluation of $f_{X_n|X_{n-1}}(x_n|x_{n-1};\theta)$
- `rmeasure`: a draw from $f_{Y_n|X_n}(y_n|x_n;\theta)$
- `dmeasure`: evaluation of $f_{Y_n|X_n}(y_n|x_n;\theta)$
- `rinit`: a draw from $f_{X_0}(x_0;\theta)$

These basic model components define the specific POMP model under consideration.

What is a simulation-based method?

- Simulating random processes is often much easier than evaluating their transition probabilities.
- In other words, we may be able to write rprocess but not dprocess.
- *Simulation-based* methods require the user to specify rprocess but not dprocess.
- *Plug-and-play*, *likelihood-free* and *equation-free* are alternative terms for "simulation-based'' methods.
- Much development of simulation-based statistical methodology has occurred in the past decade.

The `pomp` package

The `pomp` package for POMP models

- `pomp` is an Rpackage for data analysis using partially observed Markov process (POMP) models (King et al., 2016).
- Note the distinction: lower case `pomp` is a software package; upper case POMP is a class of models.
- `pomp` builds methodology for POMP models in terms of arbitrary user-specified POMP models.
- `pomp` provides tools, documentation, and examples to help users specify POMP models.
- `pomp` provides a platform for modification and sharing of models, data-analysis workflows, and methodological development.

Structure of the `pomp` package

It is useful to divide the `pomp` package functionality into different levels:

- Basic model components
- Workhorses
- Elementary POMP algorithms
- Inference algorithms

Basic model components

Basic model components are user-specified procedures that perform the elementary computations that specify a POMP model.
There are nine of these:

- `rinit`: simulator for the initial-state distribution, i.e., the distribution of the latent state at time t_0.
- `rprocess` and `dprocess`: simulator and density evaluation procedure, respectively, for the process model.
- `rmeasure` and `dmeasure`: simulator and density evaluation procedure, respectively, for the measurement model.
- `rprior` and `dprior`: simulator and density evaluation procedure, respectively, for the prior distribution.
- `skeleton`: evaluation of a deterministic skeleton.
- `partrans`: parameter transformations.

The scientist must specify whichever of these basic model components are required for the algorithms that the scientist uses.

Workhorses

Workhorses are Rfunctions, built into the package, that cause the basic model component procedures to be executed.

- Each basic model component has a corresponding workhorse.
- Effectively, the workhorse is a vectorized wrapper around the basic model component.
- For example, the `rprocess()` function uses code specified by the rprocess model component, constructed via the `rprocess` argument to `pomp()`.
- The rprocess model component specifies how a single trajectory evolves at a single moment of time.

The `rprocess()` workhorse combines these computations for arbitrary collections of times and arbitrary numbers of replications.

Elementary POMP algorithms

These are algorithms that interrogate the model or the model/data confrontation without attempting to estimate parameters.
There are currently four of these:

- `simulate` performs simulations of the POMP model, i.e., it samples from the joint distribution of latent states and observables.
- `pfilter` runs a sequential Monte Carlo (particle filter) algorithm to compute the likelihood and (optionally) estimate the prediction and filtering distributions of the latent state process.
- `probe` computes one or more uni- or multi-variate summary statistics on both actual and simulated data.
- `spect` estimates the power spectral density functions for the actual and simulated data.

POMP inference algorithms

These are procedures that build on the elementary algorithms and are used for estimation of parameters and other inferential tasks.
There are currently ten of these:

- `abc`: approximate Bayesian computation
- `bsmc2`: Liu-West algorithm for Bayesian SMC
- `pmcmc`: a particle MCMC algorithm
- `mif2`: iterated filtering (IF2)
- `enkf`, `eakf` ensemble and ensemble adjusted Kalman filters
- `traj_objfun`: trajectory matching
- `spect_objfun`: power spectrum matching
- `probe_objfun`: probe matching
- `nlf_objfun`: nonlinear forecasting

Objective function methods: among the estimation algorithms just listed, four are methods that construct stateful objective functions that can be optimized using general-purpose numerical optimization algorithms such as `optim`, `subplex`, or the optimizers in the `nloptr` package.

The `pypomp` package

Structure of the `pypomp` package

Basic model components are user-specified procedures that perform the elementary computations that specify a POMP model.
There are nine of these:

- `RInit`: simulator for the initial-state distribution, i.e., the distribution of the latent state at time t_0.

- `RProc`: simulator and density evaluation procedure, respectively, for the process model.

- `RMeas` and `DMeas`: simulator and density evaluation procedure, respectively, for the measurement model.
- LG(), dacca(), spx(), UKMeasles() : Four ready-to-run example models returning a Pomp object.

The scientist must specify whichever of these basic model components are required for the algorithms that the scientist uses.

key methods of the pyomp class

- `sample_params(bounds, n, key)`: Uniformly sample parameters within given bounds.
- `simulate(key, theta=None, times=None, nsim=…)`: Simulate latent states + observations.
- `pfilter(J, key, …)`: Particle filtering (optionally with multiple repeats).
- `mif(sigmas, M, a, J, key, …)`: IF2 iterated filtering to maximise likelihood.
- `train(J, itns, key, optimizer='Newton', …)`: Optimisation based on auto-diff gradients.
- `mop(J, key, alpha=0.97, …)`: Evaluate the Monte Carlo objective.
- `traces()` / `results(idx)`: Inspect parameter / log-likelihood histories.
- `plot_traces()`: Visualise convergence and diagnostics.

References

Andrade, J. and Duggan, J. (2022). Inferring the effective reproductive number from deterministic and semi-deterministic compartmental models using incidence and mobility data. *PLoS Comput Biol*, 18(6):e1010206.

Baracchini, T., King, A. A., Bouma, M. J., Rodó, X., Bertuzzo, E., and Pascual, M. (2017). Seasonality in cholera dynamics: a rainfall-driven model explains the wide range of patterns in endemic areas. *Adv Water Resour*, 108C:357–366.

Bjørnstad, O. N. and Grenfell, B. T. (2001). Noisy clockwork: Time series analysis of population fluctuations in animals. *Science*, 293:638–643.

Blake, I. M., Martin, R., Goel, A., Khetsuriani, N., Everts, J., Wolff, C., Wassilak, S., Aylward, R. B., and Grassly, N. C. (2014). The role of older children and adults in wild poliovirus transmission. *Proc Natl Acad Sci*, 111(29):10604–10609.

Domenech de Cellès, M., Magpantay, F. M. G., King, A. A., and Rohani, P. (2018). The impact of past vaccination coverage and immunity on pertussis resurgence. *Sci Transl Med*, 10(434):eaaj1748.

Fox, S. J., Lachmann, M., Tec, M., Pasco, R., Woody, S., Du, Z., Wang, X., Ingle, T. A., Javan, E., Dahan, M., Gaither, K., Escott, M. E., Adler, S. I., Johnston, S. C., Scott, J. G., and Meyers, L. A. (2022). Real-time pandemic surveillance using hospital admissions and mobility data. *Proc Natl Acad Sci*, 119(7):e2111870119.

King, A. A., Domenech de Cellès, M., Magpantay, F. M. G., and Rohani, P. (2015). Avoidable errors in the modelling of outbreaks of emerging pathogens, with special reference to Ebola. *Proc R Soc Lond B*, 282(1806):20150347.

King, A. A., Ionides, E. L., Pascual, M., and Bouma, M. J. (2008). Inapparent infections and cholera dynamics. *Nature*, 454(7206):877–880.

King, A. A., Nguyen, D., and Ionides, E. L. (2016). Statistical inference for partially observed Markov processes via the R package pomp. *J Stat Softw*, 69(12):1–43.

Laneri, K., Bhadra, A., Ionides, E. L., Bouma, M., Dhiman, R. C., Yadav, R. S., and Pascual, M. (2010). Forcing versus feedback: epidemic malaria and monsoon rains in northwest India. *PLoS Comput Biol*, 6(9):e1000898.

Lavine, J. S., King, A. A., Andreasen, V., and Bjørnstad, O. N. (2013). Immune boosting explains regime-shifts in prevaccine-era pertussis dynamics. *PLoS ONE*, 8(8):e72086.

Pons-Salort, M. and Grassly, N. C. (2018). Serotype-specific immunity explains the incidence of diseases caused by human enteroviruses. *Science*, 361(6404):800–803.

Romero-Severson, E. O., Volz, E., Koopman, J. S., Leitner, T., and Ionides, E. L. (2015). Dynamic variation in sexual contact rates in a cohort of HIV-negative gay men. *Am J Epidemiol*, 182(3):255–262.

Shah, M., Ferra, G., Fitzgerald, S., Barreira, P. J., Sabeti, P. C., and Colubri, A. (2022). Containing the spread of mumps on college campuses. *R Soc Open Sci*, 9(1):210948.

Shirreff, G., Zahar, J.-R., Cauchemez, S., Temime, L., and Opatowski, L. (2022). Measuring basic reproduction number to assess effects of nonpharmaceutical interventions on nosocomial SARS-CoV-2 transmission. *Emerg Infect Dis*, 28(7):1345–1354.

Subramanian, R., He, Q., and Pascual, M. (2021). Quantifying asymptomatic infection and transmission of COVID-19 in New York City using observed cases, serology, and testing capacity. *Proc Natl Acad Sci*, 118(9):e2019716118.