Bridging Statistics and Dynamic Modeling with Vensim, Python, and Stan

Tom Fiddaman, CTO, Ventana systems
tom@ventanasystems.com
Angie H. Moon, Ph.D. student, MIT
amoon@mit.edu

Thanks to Hazhir Rahmandad and students of MIT Sloan 15.879 for useful comments and contributions to this work.

Abstract

Stanify is a new library for translating Vensim models into probabilistic programs defined with the Stan language. It is composed of a source-to-source code translator for model conversion and a miniature modeling language for specifying Bayesian models on top of the Vensim model.

Stanify brings together the powerful, intuitive interface for designing dynamic models of Vensim and the robust inference performance of Stan. Stanify allow users to specify priors for Vensim variables, declare observational models on top of their existing Vensim model without having to write any Stan code. Stan provides access to different algorithms, including Hamiltonian Monte Carlo, and diverse diagnostics.

The workflow includes two novelties: symmetry between generator and estimator can be leveraged to graphically validate procedures from end to end, including the model, priors, and methods; samples from the estimation can be reinjected to Vensim sensitivity simulations to support decision making under uncertainty.

We introduce these methods with an example, exploring variants of the classic Lotka-Volterra predator-prey model, using Bayesian hierarchy and simulation-based calibration (SBC).

Workflow and Tools

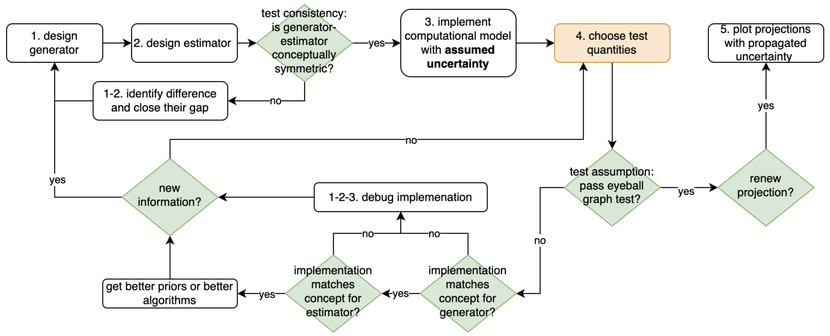


Fig.2 Testing workflow. For detailed guidelines for the workflow, especially on test statistics choice (step 4), refer to simulation-based calibration paper by Modrák et al. (2023).

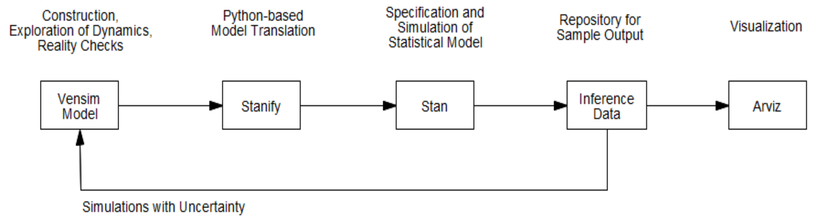


Fig.3 Architecture of Vensim + Stan on Python. Stanify exploits open-source tools, including PySD, to translate the Vensim model into a Stan-readable format. Stan output is then directed into an ArviZ inferenceData object for visualization.

Next Steps

1. Answering what benefit Bayesian inference can provide over the conventional inferential methods for ODE models

- in line with sequential and structured decision making, easier to test

- uncertainty modeling and control: possible to model uncertainty in any input (scalar: fractional adjustment, process and measurement noise variance, vector: driving data, function: ode equation)

2. Answering implementation difficulties of Bayesian inference for typical ODE models in SD domain and how to overcome?

degeneracy type	posterior shape	suppression strategies	e.g.
additive	line	tight prior, scaling strategies, compares the weakly informative prior model to the reparameterization and removal	two compartments and their fractional adjustment
multiplicative	hyperbola	scaling, fix one	one company's market size * market share
discrete allocation: exchangeable mixture model	multimodal	embed prior known order, convexification	allocation, leader and follower companies' market share

3. Collecting case studies introducing Bayesian analysis for ODE models

- hierarchical Bayesian epidemic and economy + Dream MCMC (Hazhir Rahmandad, Evan Feldman, Angie Moon)
- hierarchical Bayesian good job strategy + Dream MCMC (Hazhir, Jason Friedman)
- hierarchical Bayesian chronic waste disease + Dream MCMC (Tom Fiddaman)
- epidemic model + HMC (Jair Andrade, Jim Duggan)
- planetary movement + HMC (Andrew Gelman et al.)
- disease transmission + HMC (Léo Grinsztajn et al.)
- delegated improvement efforts on inter-level trust and future success + (tbd) (Cathy DiGennaro, Vicky Yang)

4. Identifying common failure modes in Fig.2

BayesSD, Verification, Validation

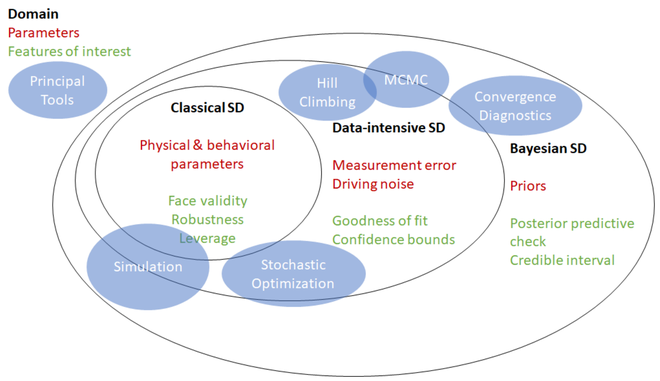
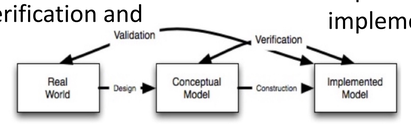
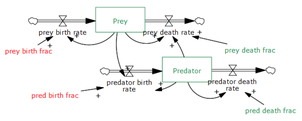


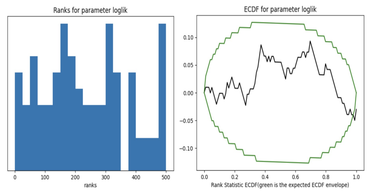
Fig.1 With increased complexity of each model components (computation, statistical, data), principled verification and validation workflow are in need.



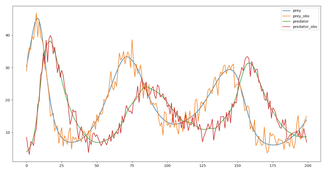
Results



From predator-prey model, birth fractional adjustment for prey and predator (in red) were estimated 500 times and compared with 500 ground truth parameter values sampled from prior distribution.



verification:
black line within oval boundary implies conceptual and implemented model matches



validation:
similar time series for generated and observed implies real world and implemented model matches

Reference

- Modrák, M., Moon, A. H., Kim, S., Bürkner, P., Huurre, N., Faltejsková, K., ... & Vehtari, A. (2022). Simulation-Based Calibration Checking for Bayesian Computation: The Choice of Test Quantities Shapes Sensitivity. arXiv preprint arXiv:2211.02383.
- Stanify library documentation, https://data4dm.github.io/stanify/stanify.html
- Vensim Data & Calibration workshops (ISDC 2022) https://vensim.com/conference/#using-data-in-vensim
- Martin-Martinez, E., Samsó, R., Houghton, J., & Solé, J. (2022). PySD: System Dynamics Modeling in Python. Journal of Open Source Software. https://doi.org/10.5281/zenodo.6583528

More …