Journal Club
Oct 5, 2021

Jakob Schlör

Universität Tübingen

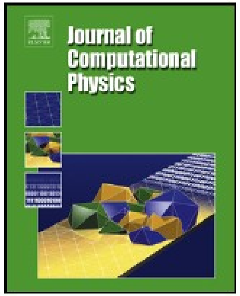
machine learning in climate science

Oct 5, 2021

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp



Calibrate, emulate, sample

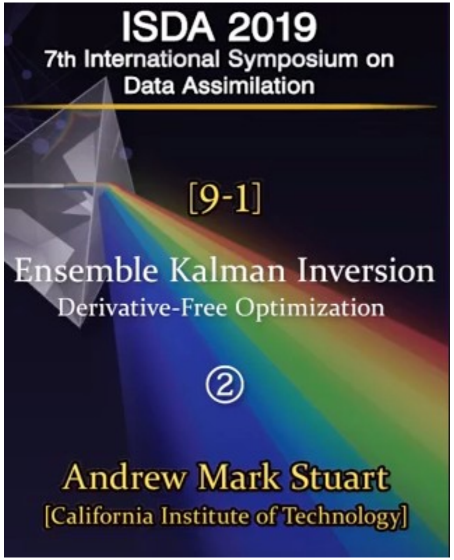
Emmet Cleary [a], Alfredo Garbuno-Inigo [a,*], Shiwei Lan [b], Tapio Schneider [a], Andrew M. Stuart [a]

[a] *California Institute of Technology, Pasadena, CA, United States of America*
[b] *Arizona State University, Tempe, AZ, United States of America*

Published: July 8, 2020

Online talks

- AI FOR GOOD DISCOVERY: AI-Accelerated Climate Modeling by Tapio Schneider at Caltech
- ISDA 2019: Ensemble Kalman inversion derivative free-optimization 1 by Andrew M. Stuart
- ISDA 2019: Ensemble Kalman inversion derivative free-optimization 2 by Andrew M. Stuart
- Dynamics Days 2020: Ensemble Kalman Inversion As A Dynamical System - Andrew Stuart



In short:

Question:

Model parameter estimation and uncertainty quantification from observational data

Results:

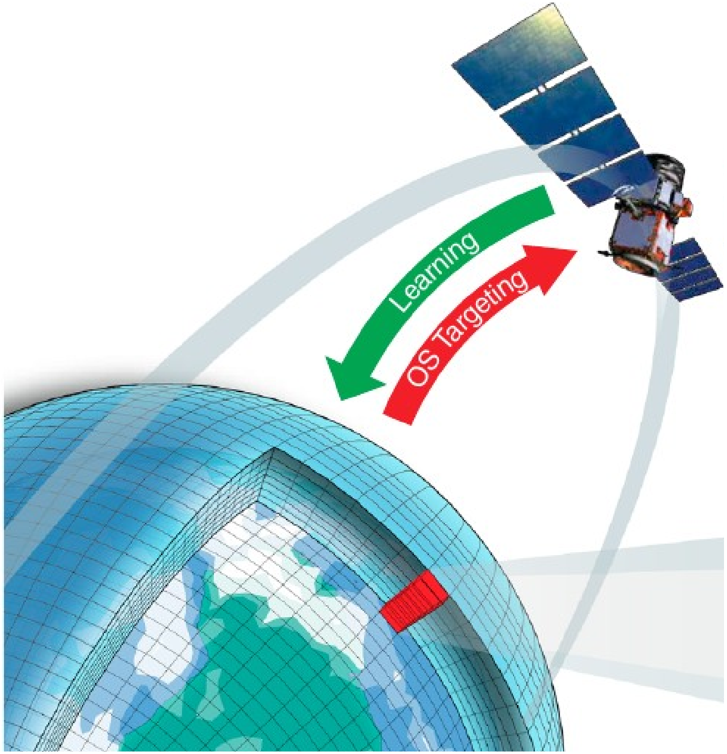
Using Kalman inversion, Gaussian process emulation and Bayesian inference allows parameter estimation of expensive to evaluate models

Impact:

Improve prediction ability and uncertainty estimation of Global Circulation Models (GCMs)

Motivation

- Climate models require parameter estimation based on observational data
- Noisy data requires uncertainty estimation of parameters
- Typical Bayesian inversion methods (MCMC) are not feasible



Schneider et al., GRL (2017)

Bayesian Inversion

Find θ from y where $G : \mathcal{U} \mapsto \mathcal{Y}, \eta$ is noise and

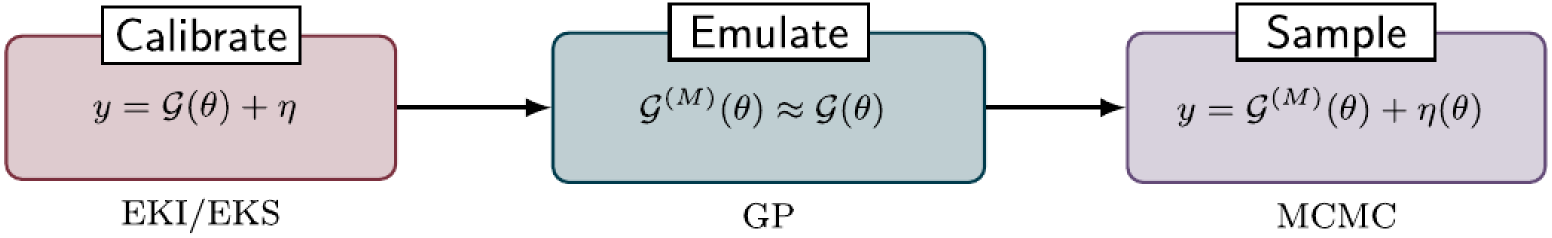
$$\mathrm{y{=}G}(\theta) + \eta$$

with $\eta \sim \mathrm{N}(0, \Gamma_y), \theta \sim \mathrm{N}(0, \Gamma_\theta)$

i.e. estimate the posterior distribution:

$$p(\theta|y) = \pi^y(\theta)$$

Method: Calibrate-Emulate-Sample (ECS)

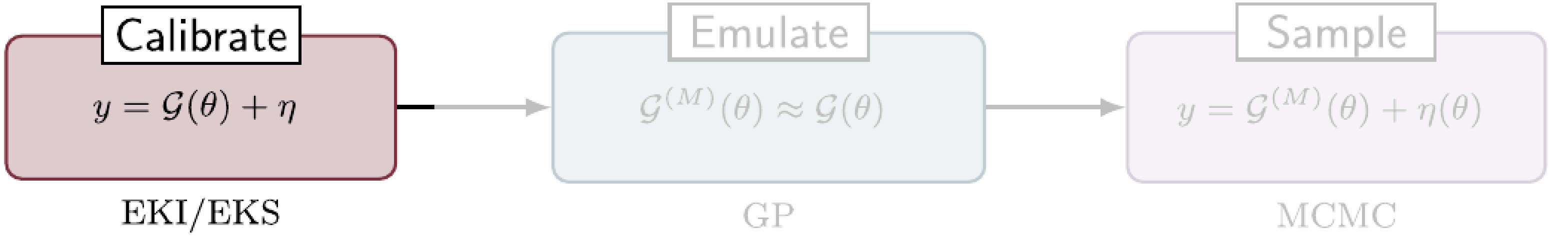


1) Calibrate: Ensemble Kalman sampling (EKS) for experimental design

2) Emulate: Gaussian Process (GP) to emulate parameter-data map

3) Sample: Markov Chain Monte Carlo (MCMC) sampling for posterior estimation

Method: Calibrate-Emulate-Sample



1) Calibrate: Ensemble Kalman sampling (EKS) for experimental design

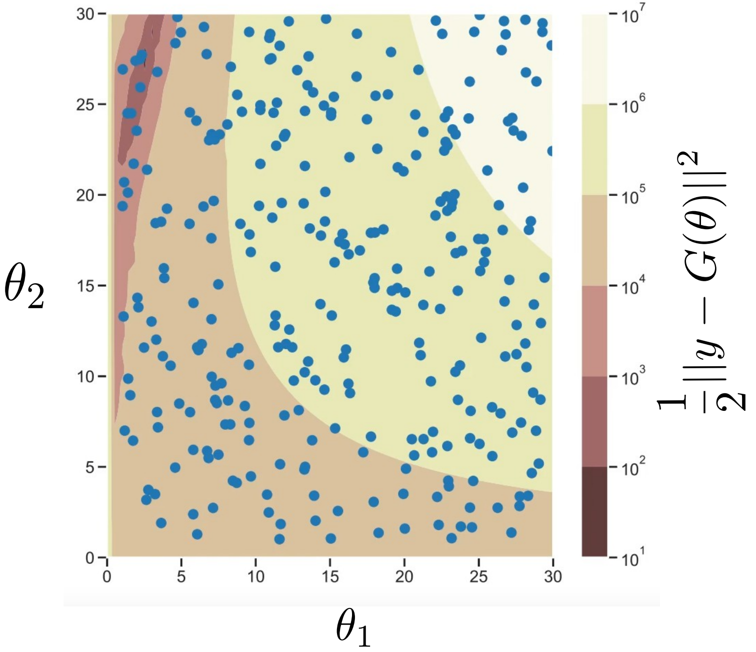
2) Emulate: Gaussian Process (GP) to emulate parameter-data map

3) Sample: Markov Chain Monte Carlo (MCMC) sampling for posterior estimation

1. Ensemble Kalman Inversion

Data assimilation method for state estimation of noisily time-dependent problems.

- Derivative-free optimization
- Iterative
- NJ model evaluations required
  (N: num. of iterations, J: num. of particles)



Update procedure:

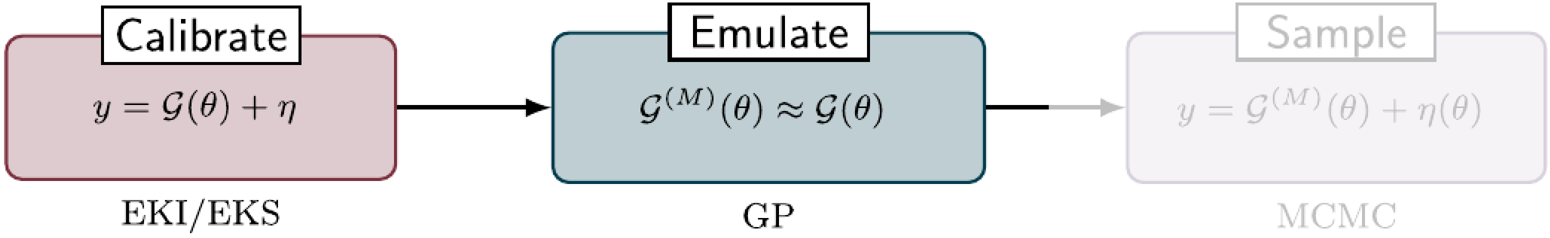
$$\frac{d\theta^{(j)}}{dt} = -\frac{1}{J}\sum_{k=1}^{J}\left\langle \mathcal{G}\left(\theta^{(k)}\right) - \overline{\mathcal{G}}, \mathcal{G}\left(\theta^{(j)}\right) - y\right\rangle_{\Gamma_y}\left(\theta^{(k)} - \bar{\theta}\right)$$

Output pairs drawn from approx. posterior:

$$\left\{\theta_n^{(i)}, G(\theta_n^{(i)})\right\}_{i=1}^{J}$$

Experimental design

Method: Calibrate-Emulate-Sample



1) Calibrate: Ensemble Kalman sampling (EKS) for experimental design

2) Emulate: Gaussian Process (GP) to emulate parameter-data map

3) Sample: Markov Chain Monte Carlo (MCMC) sampling for posterior estimation

Emulate: Gaussian Process (GP)

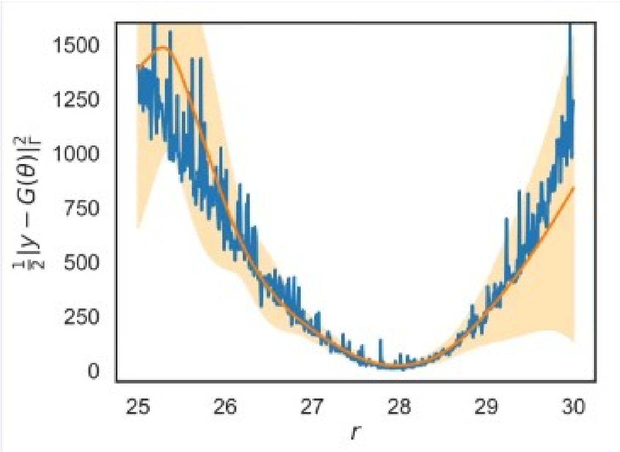
Approximate expensive-to-evaluate model G

$$y = \mathcal{G}^{(M)}(\theta) + \eta$$

by Gaussian process:

$$\mathcal{G}^{(M)}(\theta) \sim \mathrm{N}\left(m(\theta), \Gamma_{\mathrm{GP}}(\theta)\right)$$

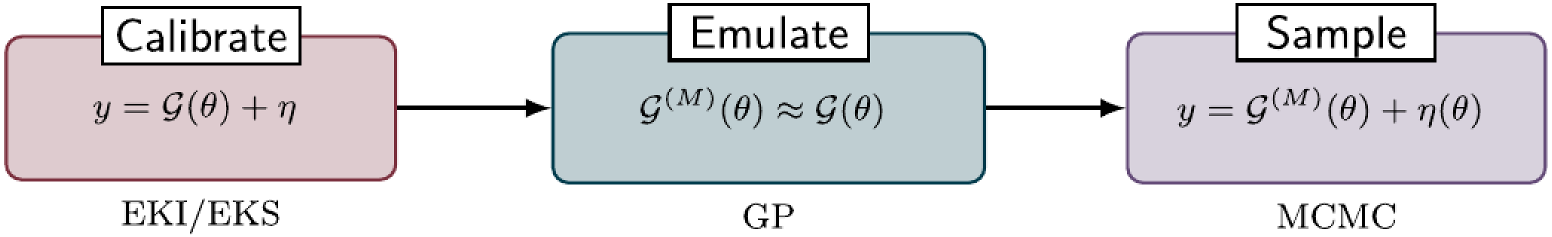
$$\text{with } k_l\left(\theta, \theta'\right) = \sigma_l^2 \exp\left(-\frac{1}{2}\left\|\theta - \theta'\right\|_{D_l}^2\right) + \lambda_l^2 \delta_\theta\left(\theta'\right)$$



Example of the Lorenz 63 model
https://www.youtube.com/watch?v=qSrfb9Dy6e4

The GP is trained on the input-output pairs from the last EKS iteration $\left\{\theta_n^{(i)}, G(\theta_n^{(i)})\right\}_{i=1}^{J}$

Method: Calibrate-Emulate-Sample



1) Calibrate: Ensemble Kalman sampling (EKS) for experimental design

2) Emulate: Gaussian Process (GP) to emulate parameter-data map

3) Sample: Markov Chain Monte Carlo (MCMC) sampling for posterior estimation

Sample - MCMC

Estimation of model output distribution based on MCMC sampling of θ

Algorithm:

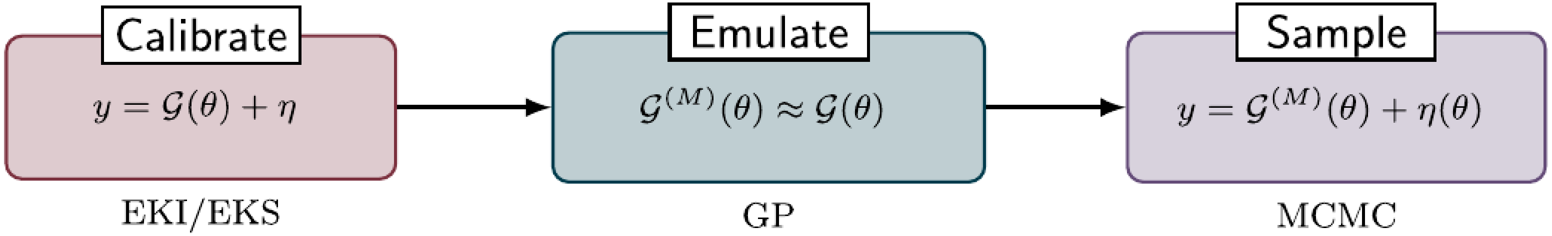
1. Choose $\theta_0 = \theta_J$
2. Propose a new pareameter choice $\theta^*_{n+1} = \theta_n + \zeta_n$
3. Set $\theta_{n+1} = \theta^*_{n+1}$ with probability $a(\theta_n, \theta^*_{n+1})$
4. $n \to n+1$, return to 2

Acceptance probability:

$$a(\theta, \theta^*) = \min\left\{1, \exp\left[\left(\Phi^{(M)}(\theta^*) + \frac{1}{2}\|\theta^*\|^2_{\Gamma_\theta}\right) - \left(\Phi^{(M)}(\theta) + \frac{1}{2}\|\theta\|^2_{\Gamma_\theta}\right)\right]\right\}$$

$$\text{with } \Phi^{(M)}_{\mathrm{GP}}(\theta) = \frac{1}{2}\|y - m(\theta)\|^2_{\Gamma_{\mathrm{GP}}(\theta)+\Gamma_y} + \frac{1}{2}\log\det\left(\Gamma_{\mathrm{GP}}(\theta) + \Gamma_y\right)$$

Method: Calibrate-Emulate-Sample

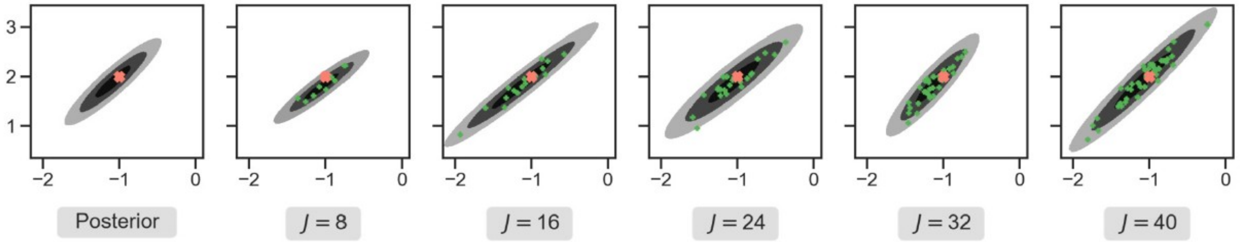


1) Calibrate: Ensemble Kalman sampling (EKS) for experimental design
2) Emulate: Gaussian Process (GP) to emulate parameter-data map
3) Sample: Markov Chain Monte Carlo (MCMC) sampling for posterior estimation

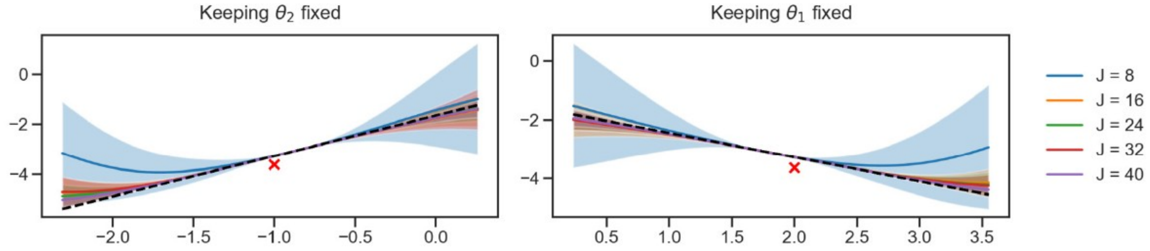
Example: Linear problem

Example on a 2-dimensional linear model: $y = G\theta^{\dagger} + \eta$

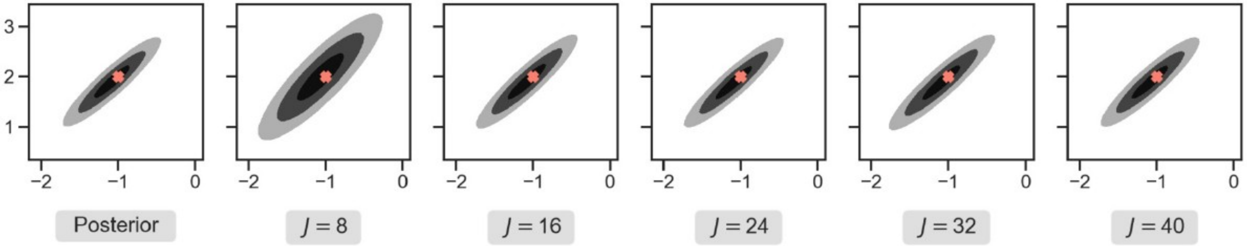
1) Calibrate: 20 iteration of EKS with J particles



2) Emulate: Train GP on EKS of last iteration



3) Sampling: MCMC to apptoximate posterior



Time-average data

In chaotic dynamical systems data may only be available in time-averaged form.

Dynamical system

$$\dot{z} = F(z;\theta),$$
$$z(0) = z_0$$

Time - average

$$\mathcal{G}_\tau(\theta; z_0) = \frac{1}{\tau}\int_{T_0}^{T_0+\tau} \varphi(z(t;\theta))dt$$

$$y = \mathrm{G}_\tau(\theta, z_0) + \eta$$
$$= \mathrm{G}_\tau(\theta) + \eta$$

Assumption: Ergodicity

Initial conditions play no role in time averages over the infinite time horizon, i.e. initial condition lead to random errors for finite times.

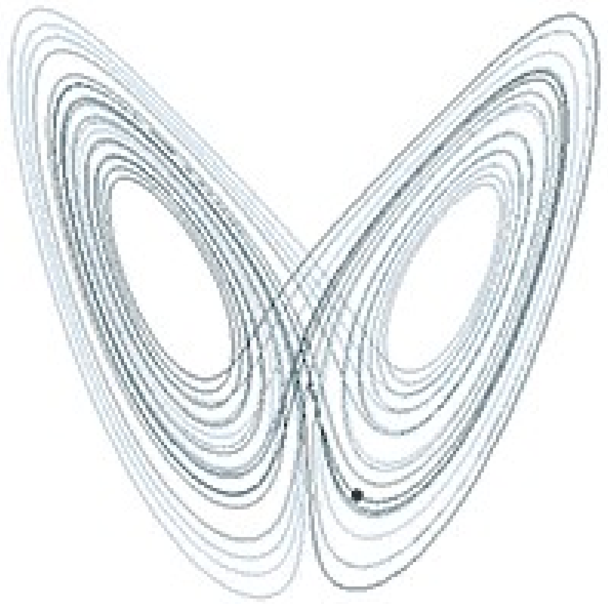
Example: Lorenz 63

Dynamical system:

$$\dot{x} = 10(y - x)$$
$$\dot{y} = rx - y - xz$$
$$\dot{z} = xy - bz$$

Parameter:

$$\theta = [r, b]^\top$$



https://en.wikipedia.org/wiki/Lorenz_system

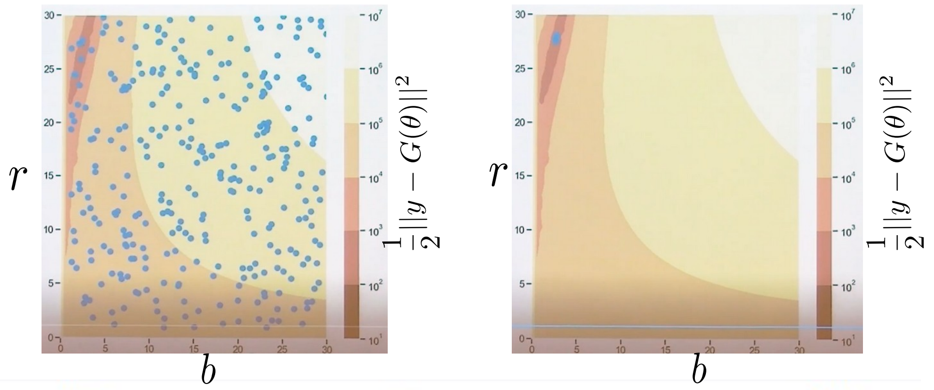
Example: Lorenz 63

Dynamical system:

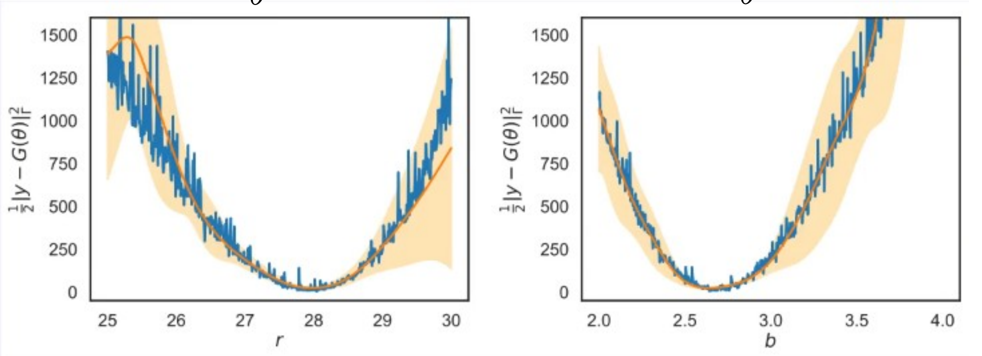
$$\dot{x} = 10(y - x)$$
$$\dot{y} = rx - y - xz$$
$$\dot{z} = xy - bz$$

Parameter: $\theta = [r, b]^\top$

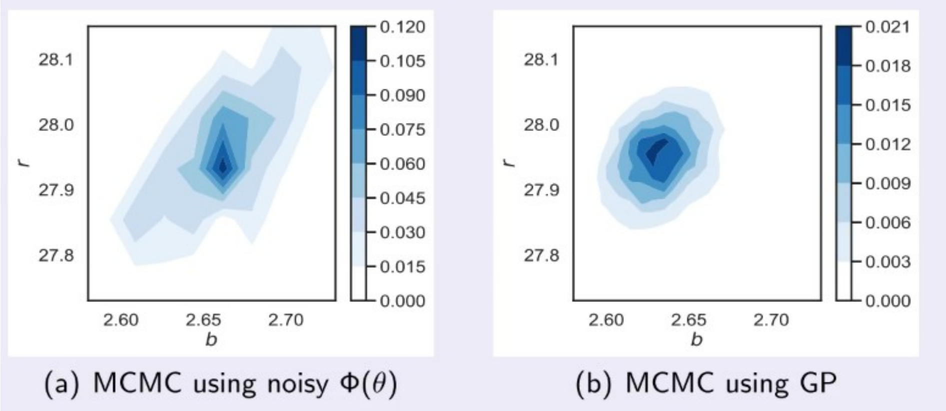
1. Calibrate



2. Emulate



3. Sample



Notes:

- MCMC on the noisy data is less efficient in terms of acceptance rate
- Less model evaluations

Example: Simplified ESM

Aqua planet ESM: Simplified Betts-Miller Scheme

$$\text{Moisture Conservation:} \quad \frac{\partial q}{\partial t} + \mathbf{v} \cdot \nabla q = -\frac{q - q_{ref}(T;\theta)}{\tau_q(q,T;\theta)}$$

$$\text{Energy Conservation:} \quad \frac{\partial T}{\partial t} + \mathbf{v} \cdot \nabla T = \frac{T - T_{ref}(q,T;\theta)}{\tau_T(q,T;\theta)} + \text{RAD} + \dots$$

https://youtu.be/qSrfb9Dy6e4?t=938

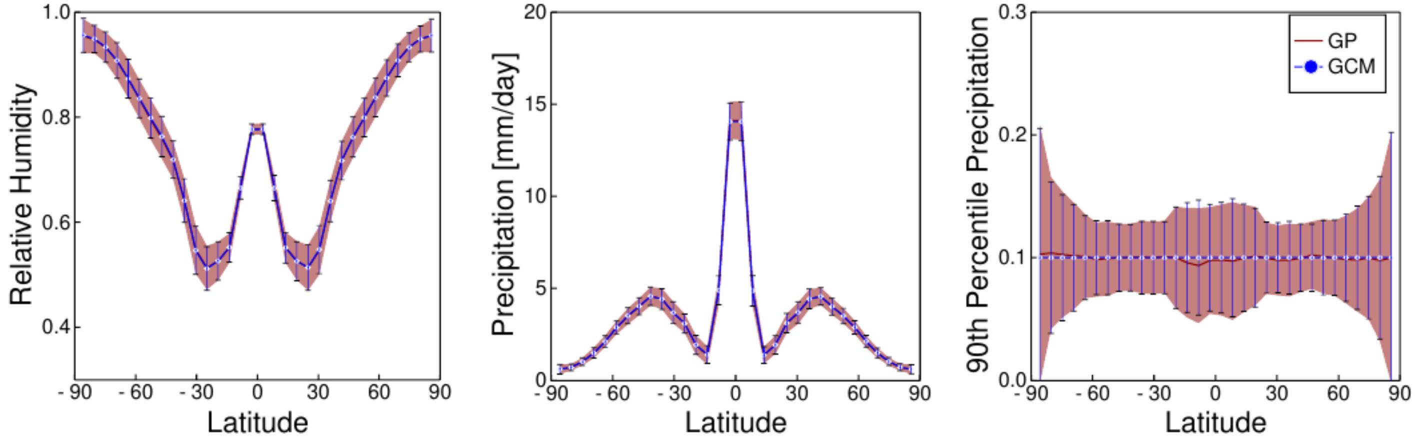


Figure 9: Comparison between the GCM statistics at the true parameters $\boldsymbol{\theta}^{\dagger}$ and the trained B-GP emulator predictions at $\boldsymbol{\theta}^{\dagger}$. Blue: GCM mean (dots) averaged over 600 30-day runs, with the error bars marking a 95% confidence interval from variances on the diagonal of Γ. Dark red: predicted mean (line) and 95% confidence interval (shaded region) produced by the B-GP emulator.

Dunbar et al. (2021)

Conclusion

Calibrate-emulate-sample approach is advantageous

- Requires modest number of runs of expensive-to-evaluate models
- Finds optimal parameters even with noisy data
- MCMC allows uncertainty estimation by evaluating the GP model

Outlook

- Replace GP by NN for high-dimensional parameter learning
- Generalize for structural model errors

Ensemble Kalman Inversion

https://youtu.be/Mm6ft793vK0?t=318

Ensemble Kalman method

State Space Model:

$$\text{Dynamics Model:} \quad v_{n+1} = \Psi\left(v_n\right) + \xi_n, \quad n \in \mathbb{Z}^+$$

$$\text{Data Model:} \quad y_{n+1} = H v_{n+1} + \eta_{n+1}, \quad n \in \mathbb{Z}^+$$

$$\text{Probabilistic Structure:} \quad v_0 \sim N\left(m_0, C_0\right), \quad \xi_n \sim N(0, \Sigma), \quad \eta_n \sim N(0, \Gamma)$$

Optimization approach:

$$\text{Predict:} \quad \widehat{v}_{n+1} = \Psi\left(v_n\right), \quad n \in \mathbb{Z}^+$$

$$\text{Model/Data Compromise:} \quad J_n(v) = \frac{1}{2}\left|v - \widehat{v}_{n+1}\right|_{\widehat{C}}^2 + \frac{1}{2}\left|y_{n+1} - H v\right|_r^2$$

$$\text{Optimize:} \quad v_{n+1} = \operatorname{argmin}_v J_n(v)$$

Inverse Problem

Problem Statement

Find u from y where $\mathrm{G} : \mathcal{U} \mapsto \mathcal{Y}$, η is noise and

$$y = \mathrm{G}(u) + \eta.$$

Optimization $\Phi(u) = \frac{1}{2}|y - G(u)|_\Gamma^2 + \frac{1}{2}|u|_\Sigma^2;$ Probability $e^{-\Phi(u)}.$

Dynamical Formulation

Iterative inversion: see [9], [11], [16]

$$\begin{aligned}
\text{Dynamics Model:} \quad & u_{n+1} = u_n, \quad n \in \mathbb{Z}^+ \\
\text{Dynamics Model:} \quad & w_{n+1} = \mathrm{G}(u_n), \quad n \in \mathbb{Z}^+ \\
\text{Data Model:} \quad & y_{n+1} = w_{n+1} + \eta_{n+1}, \quad n \in \mathbb{Z}^+
\end{aligned}$$

https://youtu.be/Mm6ft793vK0

Inverse Problem

Dynamical Formulation

$$\text{Dynamics Model:} \quad u_{n+1} = u_n, \quad n \in \mathbb{Z}^+$$

$$\text{Dynamics Model:} \quad w_{n+1} = \mathsf{G}(u_n), \quad n \in \mathbb{Z}^+$$

$$\text{Data Model:} \quad y_{n+1} = w_{n+1} + \eta_{n+1}, \quad n \in \mathbb{Z}^+$$

State Space Estimation Formulation

$$\text{Reformulate:} \quad v = (u, w), \quad \Psi(v) = (u, \mathsf{G}(u)), \quad H = (0, I)$$

$$\text{Dynamics Model:} \quad v_{n+1} = \Psi(v_n), \quad n \in \mathbb{Z}^+$$

$$\text{Data Model:} \quad y_{n+1} = Hv_{n+1} + \eta_{n+1}, \quad n \in \mathbb{Z}^+$$

Employ EnKF with $y_{n+1} \equiv y$.

https://youtu.be/Mm6ft793vK0

Inverse Problem

Iteration $n \mapsto n+1$

$$u_{n+1}^{(j)} = u_n^{(j)} + C_n^{uw}\left(C_n^{ww} + \Gamma\right)^{-1}\left(y - G(u_n^{(j)})\right)$$

https://youtu.be/Mm6ft793vK0