Learning Hyperparameters using Bayesian Optimization with Optimized Precision

Mattias Villani

Department of Statistics
Stockholm University

mattiasvillani.com @matvil @matvil mattiasvillani

Overview

- **Hyperparameter learning**
- **Gaussian processes** and **Bayesian optimization**
- **BOOP: Bayesian Optimization with Optimized Precision**
- **Applications in Econometrics**

- **Slides**: http://mattiasvillani.com/news

Joint work with

- **Oskar Gustafsson**, Dept of Statistics, Stockholm University
- **Pär Stockhammar**, Sveriges Riksbank

Parameters and Hyperparameters - examples

- Distinction:
  - **Parameters**, $\boldsymbol{\beta}$, typically top level, high-dim.
  - **Hyperparameters** $\boldsymbol{\theta}$, typically low level, low-dim.

- Bayesian **vector autoregressive models** (**VAR**) models

$$\boldsymbol{y}_t = \boldsymbol{\mu} + \sum_{k=1}^{K} \boldsymbol{A}_k (y_{t-k} - \boldsymbol{\mu}) + \boldsymbol{\varepsilon}_t, \quad \boldsymbol{\varepsilon}_t \overset{\text{iid}}{\sim} N(\mathbf{0}, \boldsymbol{\Sigma})$$

Hyperparameters $\boldsymbol{\theta} = (\lambda_1, \lambda_2, \lambda_3)$. Prior $\mathrm{Std}(A_{ij}^{(k)}) = \frac{\lambda_1 \lambda_2}{k^{\lambda_3}}$.

- **State-space models**

$$y_t = h(x_t) + \varepsilon_t, \quad \varepsilon_t \overset{\text{iid}}{\sim} N(0, \sigma_\varepsilon^2)$$
$$x_t = g(x_{t-1}) + \nu_t \quad \nu_t \overset{\text{iid}}{\sim} N(0, \sigma_\nu^2)$$

- **DSGE**. $\boldsymbol{\beta}$ persistence/variance of shocks, $\boldsymbol{\theta}$ steady state.

- **Deep neural net**: $\boldsymbol{\beta}$ are weights, $\boldsymbol{\theta}$ is network architecture.

Hyperparameter optimization

- Practitioners prefer to fix $\boldsymbol{\theta}$ "once and for all". Move on to parameter inference, model checking, forecasting, policy etc
- BVARs: "... use the hyperparameters from Doan et al (1984)"
- **Maximize marginal likelihood**

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \log p(\boldsymbol{Y}_{1:T}|\boldsymbol{\theta})$$

- **Empirical Bayes: maximize marginal posterior** $p(\boldsymbol{\theta}|\boldsymbol{Y}_{1:T})$

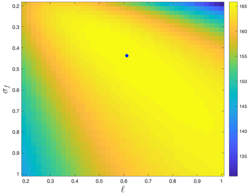
$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \log p(\boldsymbol{Y}_{1:T}|\boldsymbol{\theta}) + \log p(\boldsymbol{\theta})$$

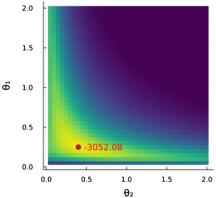
22-variable steady-state BVAR

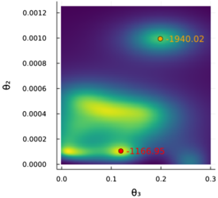
	Standard	BO-EI	BOOP-EI	Medium BVAR
Log ML	−7576.31	−7402.50	−7401.09	−7532.61
Sd log ML	0.54	0.81	0.16	0.49
Gibbs iterations		3.75×10^6	1.8×10^6	
CPU time (h)		64.90	20.22	
θ_1	0.1	0.47	0.56	0.27
θ_2	0.5	0.06	0.05	0.41
θ_3	1	1.46	1.51	0.76

Hyperparameters - It’s complicated

- **Weakly identified - flat regions**
- **Weakly identified - ridges**
- **Multimodal**







Hyperparameter optimization is tricky

- Marginal likelihood often **intractable**:
  - analytical approximation (Laplace, INLA, Variational inference)
  - HMC/MCMC simulation to compute $p(\boldsymbol{Y}_{1:T}|\boldsymbol{\theta})$.
- Typical hyperparameter optimization setup:
  - **costly** function evaluations
  - **noisy** function evaluations (marginal likelihood from MCMC)
  - function argument is **low-dimensional**.
- **Bayesian optimization** well suited for all three issues.
- Treats the underlying function as **unknown** and puts a **Gaussian process prior** on it. **Bayesian numerics**.

Gaussian processes regression

- **Gaussian process regression**

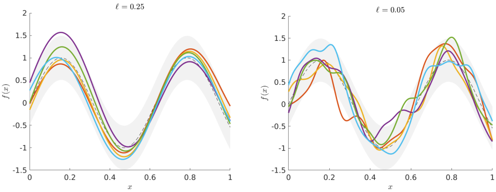
$$y_i = f(\mathbf{x}_i) + \epsilon_i, \qquad \epsilon_i \overset{\text{iid}}{\sim} N(0, \sigma_n^2)$$

- **Gaussian process prior** over the space of functions

$$f(\boldsymbol{x}) \sim \mathcal{GP}\left(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}')\right)$$

- **Squared exponential** covariance function

$$k(\boldsymbol{x}, \boldsymbol{x}') \equiv \mathrm{Cov}(f(\mathbf{x}), f(\mathbf{x}')) = \sigma_f^2 \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{2\ell^2}\right)$$



- **Posterior** of $f(x)$ is also a Gaussian process.

Bayesian optimization

- Aim: **maximization of expensive function**

$$\text{argmax}_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x})$$

- **Bayesian optimization**:
  - Assume $f \sim \mathcal{GP}$
  - Evaluate f at $x_1, x_2, ..., x_n$.
  - Update to posterior distribution $f|x_1, ..., x_n \sim \mathcal{GP}$.
  - Use posterior of f to find a new x_{n+1}.
  - Iterate until convergence.

- Find new x_{n+1} by optimizing an **acquisition function**.

Acquisition function

- **Probability of Improvement** (**PI**)

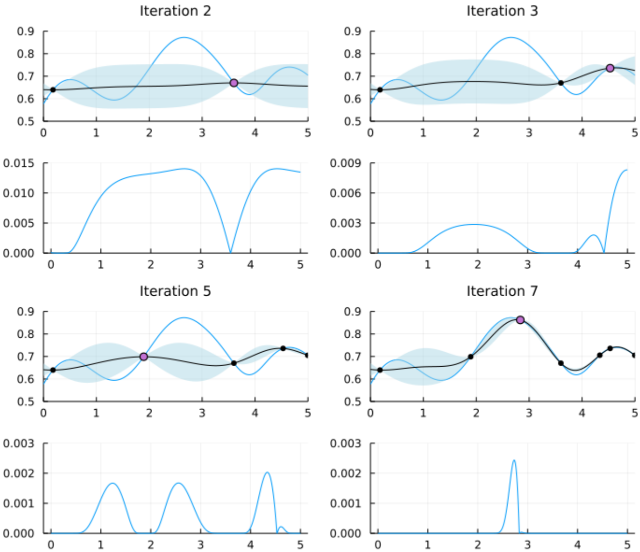
$$a(x_{n+1}) \equiv \Pr\left(f(x_{n+1}) > \max(f_{1:n}) \mid f_{1:n}\right)$$

- **Expected Improvement** (**EI**) takes also into account the **size of the improvement**.

- **Expected Improvement per Second** - takes a known function evaluation **cost** into account.

- **Non-convex** acquisition function **optimization**, but deterministic and cheaper than original problem. **Particle swarm optimization**.

BO - expected improvement



Marginal likelihood estimated from sampling

- **Marginal likelihood** $f(\boldsymbol{\theta}) \equiv \log p(\boldsymbol{Y}_{1:T}|\boldsymbol{\theta})$ estimated by:
  - Chib (Gibbs) and Chib-Jeliazkov (MH)
  - Importance sampling
  - Particle filters
- **Noisy** evaluations $\hat{f}(\boldsymbol{\theta})$.
- **Precision** of $\hat{f}(\boldsymbol{\theta})$ controlled via **number of samples G**.
- **Sampling efficiency**, $\mathbb{V}(\hat{f}(\boldsymbol{\theta}))$ **varies over θ-space**.
- **Stopping early** when probability of improvement (PI) is low.

Bayesian Optimization with Optimized Precision

- **Early stopping of evaluation** when **PI** $< \alpha$.
- **EI per second**, but with G **predicted** for every $\boldsymbol{\theta}$.
- **BOOP acquisition function** from baseline $a(\boldsymbol{x})$ (e.g. EI):

$$\tilde{a}_\alpha(\boldsymbol{x}) = \frac{a(\boldsymbol{x})}{\hat{G}_\alpha(\boldsymbol{x})}$$

- Early stopping affects the **planning of future computations**.
- BOOP can try $\boldsymbol{\theta}$ with low EI, if expected to be cheap.
- **Heteroscedastic GP regression model** for the estimates

$$\hat{f}(\boldsymbol{\theta}_i) = f(\boldsymbol{\theta}_i) + \epsilon_i, \qquad \epsilon_i \overset{\text{iid}}{\sim} N\left(0, \sigma^2(G_i)\right)$$

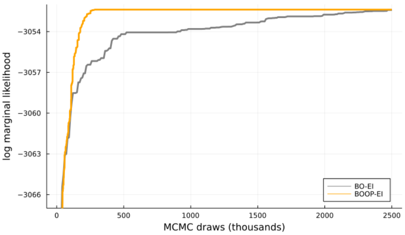
- GP for **predicting the number of samples** G:

$$\ln G_i = h(\boldsymbol{z}_i) + \varepsilon_i \qquad \varepsilon_i \overset{\text{iid}}{\sim} N\left(0, \psi^2\right),$$

where $\boldsymbol{z}$ are variables with predictive power for G.

7-variable Steady-state BVAR

- 7 variable **steady-state BVAR** on US data.
- Gibbs sampling with **Chib's marginal likelihood estimator**.
- BO to find optimal prior hyperparameters $\boldsymbol{\theta} = (\lambda_1, \lambda_2, \lambda_3)$.



7-variable BVAR - true ML surface vs predicted

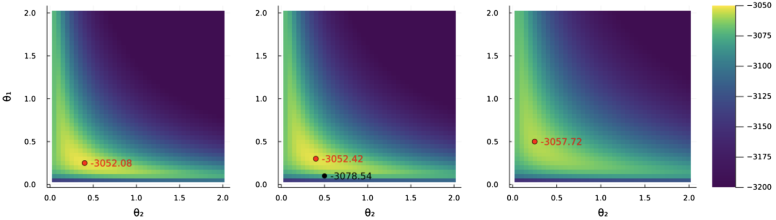


FIGURE 9 Log marginal likelihood surfaces over a fine grid of (θ_1, θ_2) values. The hyperparameter values for the lag decay are (a) $\theta_3 = 0.76$, (b) $\theta_3 = 1$, and (c) $\theta_3 = 2$ (left to right). The red dot denotes the maximum log marginal likelihood value for the given θ_3, and the black dot, in the middle plot, shows the standard values.

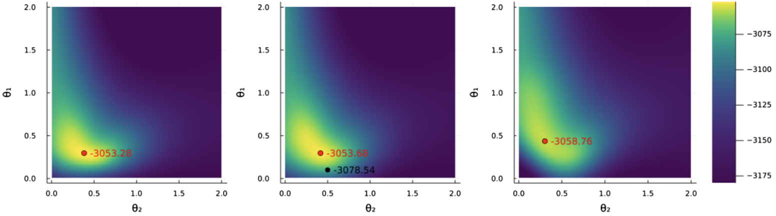


FIGURE 10 GP predictions of the hyperparameter surfaces in Figure 9 based on 250 evaluations for one BOOP-EI run. The hyperparameter for the lag decay is $\theta_3 = 0.76$, 1, and 2 (left to right). Red dot indicates the highest predicted value in the subplot, and the black dot, in the middle plot, shows the standard values.

22-variable steady-state BVAR

	Standard	BO-EI	BOOP-EI	Medium BVAR
Log ML	−7576.31	−7402.50	−7401.09	−7532.61
Sd log ML	0.54	0.81	0.16	0.49
Gibbs iterations		3.75×10^6	1.8×10^6	
CPU time (h)		64.90	20.22	
θ_1	0.1	0.47	0.56	0.27
θ_2	0.5	0.06	0.05	0.41
θ_3	1	1.46	1.51	0.76

TVP-SV BVAR [Chan and Eisenstat (2018, JAE)]

- Time-varying parameter stochastic volatility BVAR:

$$A_{0,t} y_t = c_t + \sum_{k=1}^{K} A_{k,t} y_{t-k} + \varepsilon_t, \quad \varepsilon_t \overset{\text{iid}}{\sim} N(0, \Sigma_t)$$

- Random walk evolution of $A_{k,t}$ and log variances.
- Three hyperparameters: prior mean of innovation variances $(c_t, A_t$ and $\Sigma_t)$.
- **Marginal likelihood** estimated by costly IS^2-type algorithm.

	CE	BO1	BO2	BO3	BOOP1	BOOP2	BOOP3
Log ML	−1180.2	−1169.25	−1170.57	−1178.34	−1167.32	−1172.92	−1168.49
SE	0.12	0.89	0.49	0.32	1.24	0.47	1.60
$\theta_1 \times 10^3$	40	19.05	8.66	29.53	7.65	12.22	15.14
$\theta_2 \times 10^5$	40	9.81	10.65	11.07	10.26	7.06	8.70
$\theta_3 \times 10^3$	40	77.56	119.07	25.04	73.81	25.12	114.42
Iterations	-	67	35	46	81	44	157
CPU time (h)	-	83.40	42.47	56.25	34.90	22.49	77.89

TVP-SV BVAR

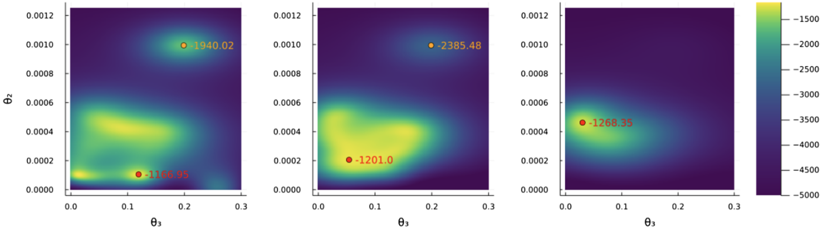


FIGURE 12 Predicted log marginal likelihood over the hyperparameters for stochastic volatility and the VAR dynamics for $\theta_1 = 0.0086$ (left), 0.05 (middle), and 0.1 (right). The mode in each plot is marked out by a red point. A distant local optimum is also marked out by an orange point.

Conclusions

- **Bayesian optimization** is an attractive method for **costly**, **noisy**, **low-dimensional functions**.
- Hyperparameter optimization using **marginal likelihood estimated from MC sampling**.
- We extend BO to exploit that **the user controls the precision of the evaluations** via the number of samples.
- Successful applications to steady-state and TVP-SV BVARs.
- **Current work**: applications to particle methods for challenging nonlinear and non-Gaussian state space models.