Bayesian Hyperparameter Learning

Mattias Villani

Department of Statistics
Stockholm University

Overview

- Hyperparameter inference - motivation
- Gaussian processes - background
- Bayesian optimization
- Bayesian Optimization with Optimized Precision
- Applications in Econometrics
- Variational inference for speeding up hyperparameter learning

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

Collaborators

- Oskar Gustafsson, Department of Statistics
- Pär Stockhammar, Sveriges Riksbank

Hyperparameter inference - stats

- Parameter/Hyperparameter distinction:
  - **Parameters**, $\boldsymbol{\beta}$, typically high-dim.
  - **Hyperparameters** θ, typically low-dim.
- **L2-regularized linear regression**

$$y_i = \boldsymbol{x}_i^\top \boldsymbol{\beta} + \varepsilon_i$$

$$\beta_j | \sigma^2 \overset{\text{iid}}{\sim} N\left(0, \tau^2 \sigma^2\right)$$

- **Horseshoe-regularized linear regression**

$$\beta_j | \sigma^2 \overset{\text{iid}}{\sim} N\left(0, \tau^2 \lambda_j^2 \sigma^2\right)$$

$$\lambda_j \sim C^+(0, 1)$$

- **State-space models**

$$y_t = \mu_t + \varepsilon_t, \quad \varepsilon_t \overset{\text{iid}}{\sim} N(0, \sigma_\varepsilon^2)$$

$$\mu_t = \mu_{t-1} + \nu_t \quad \nu_t \overset{\text{iid}}{\sim} N(0, \sigma_\nu^2)$$

Hyperparameter inference - machine learning

Deep neural networks

- $\boldsymbol{\beta}$ are the weights and biases
- $\boldsymbol{\theta}$ is the network architecture
  - no. of layers
  - no. of nodes
  - filters
  - learning rate

Gaussian process regression/classification/spatial

$$y_i = f(x_i) + \varepsilon_i$$

- $\boldsymbol{\beta} = (f_1, \ldots, f_n)^\top$ is the unknown function at n test points
- $\boldsymbol{\theta}$ kernel hyperparameters determining the smoothness of $f(\cdot)$.

Hyperparameter inference - econometrics

- DSGE models in econometrics
  - $\boldsymbol{\beta}$ are the persistence and variance of shocks etc
  - $\boldsymbol{\theta}$ are parameters in the steady state.

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

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

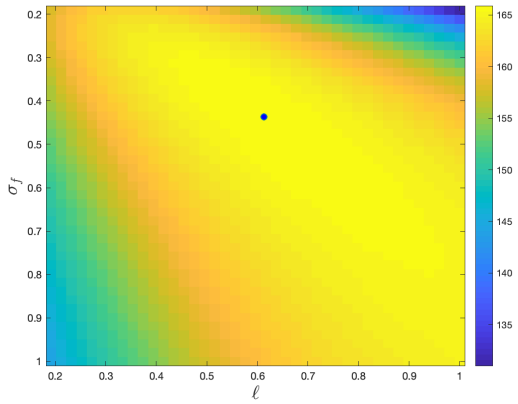
  - $\boldsymbol{\beta} = (\mu, A_1, \ldots, A_K, \Sigma)$
  - $\boldsymbol{\theta} = (\lambda_1, \lambda_2, \lambda_3)$ determine the prior standard deviation:

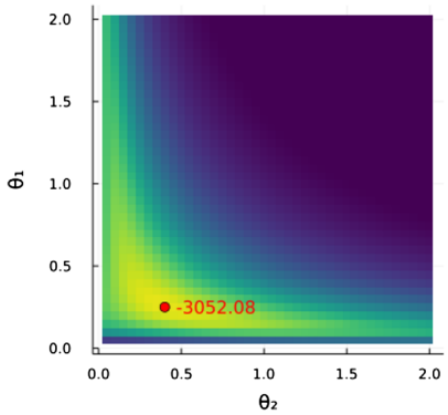
$$\text{Std}(a_{ij}^{(k)}) = \begin{cases} \frac{\lambda_1}{k^{\lambda_3}} & \text{own lags } i = j \\ \frac{\lambda_1 \lambda_2}{k^{\lambda_3}} & \text{foreign lags } i \neq j \end{cases}$$

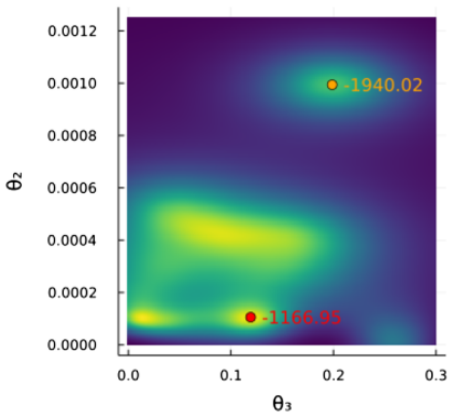
- **Time-varying VAR** with **stochastic volatility**

Hyperparameters - It's complicated

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







Hyperparameter inference - sampling

- **Computational methods for sampling** from $p(\boldsymbol{\beta}, \boldsymbol{\theta} | \boldsymbol{Y}_{1:T})$:
  - **Direct sampling** (rarely an option)
  - **MCMC/HMC** on joint $\boldsymbol{\beta}, \boldsymbol{\theta} | \boldsymbol{Y}_{1:T}$
  - **Gibbs sampling** $\boldsymbol{\beta} | \boldsymbol{\theta}, \boldsymbol{Y}_{1:T}$ and $\boldsymbol{\theta} | \boldsymbol{\beta}, \boldsymbol{Y}_{1:T}$
  - **Pseudo-marginal** samplers
- Hyperparameters $\boldsymbol{\theta}$ tend to
  - have complicated posteriors
  - correlate with the model parameters $\boldsymbol{\beta}$.
- Joint learning of parameter and hyperparameters **slows down HMC/MCMC/Gibbs convergence**.

Hyperparameter optimization

- Practitioners prefer to fix $\boldsymbol{\theta}$ "once and for all". Move on to parameter inference, model checking, forecasting, policy etc
- Bayesian VARs: "we 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** $p(\boldsymbol{\theta}|\boldsymbol{Y}_{1:T})$

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

Hyperparameter optimization is tricky

- Marginal likelihood often **intractable**:
  - analytical approximation (Laplace, INLA, Variational inference)
  - HMC/MCMC simulation to compute $p(\boldsymbol{\theta}|\boldsymbol{Y}_{1:T})$.
- 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**, Probabilistic numerics.

Gaussian processes regression

- **Gaussian process regression**

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

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

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

- The covariance between any two function ordinates is expressed with a **covariance kernel**,

$$k(\boldsymbol{x}, \boldsymbol{x}') \equiv \mathrm{Cov}(f(\mathrm{x}), f(\mathrm{x}')).$$

- **Squared exponential** covariance function

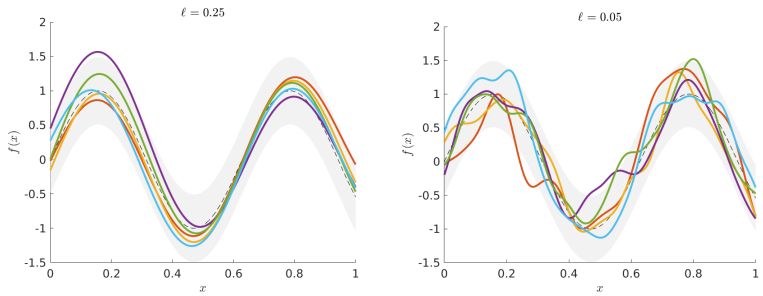
$$k(\boldsymbol{x}, \boldsymbol{x}') \equiv \mathrm{Cov}(f(\mathrm{x}), f(\mathrm{x}')) = \sigma_f^2 \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{2\ell^2}\right)$$

Nearby inputs x have correlated f(x)

- **Squared exponential** covariance function

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

- The **correlation decays with** $\|\boldsymbol{x} - \boldsymbol{x}'\|$ depending on length scale ℓ.



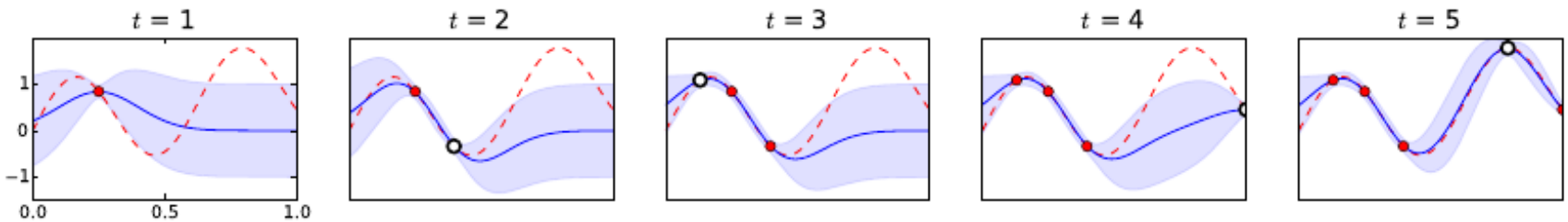
- The variance around the mean function is given by σ_f^2.
- **Matern5/2 kernel** which has two continuous MS derivatives.

Gaussian processes regression posterior

- **Posterior** of $f(\cdot)$ at new test point $\boldsymbol{x}_\star$ is also Gaussian

$$\begin{aligned} \mathrm{f}_* | \boldsymbol{X}, \mathrm{y}, \boldsymbol{x}_* &\sim N\left(\bar{\mathrm{f}}_*, \mathrm{cov}(\mathrm{f}_*)\right) \\ \bar{\mathrm{f}}_* &= k(\boldsymbol{x}_*, \boldsymbol{x}) \left[K(\mathrm{X}, \mathrm{X}) + \sigma_n^2 I\right]^{-1} \mathrm{y} \\ \mathrm{cov}(\mathrm{f}_*) &= k(\boldsymbol{x}_*, \boldsymbol{x}_*) - k(\boldsymbol{x}_*, \boldsymbol{x}) \left[K(\mathrm{X}, \mathrm{X}) + \sigma_n^2 I\right]^{-1} k(\boldsymbol{x}, \boldsymbol{x}_*) \end{aligned}$$

- The $n \times n$ **kernel matrix** $K(\boldsymbol{X}, \boldsymbol{X})$ is computed by applying the **kernel function** $k(\boldsymbol{x}, \boldsymbol{x}')$ to all input pairs.



Bayesian optimization

- Aim: **maximization of expensive function**

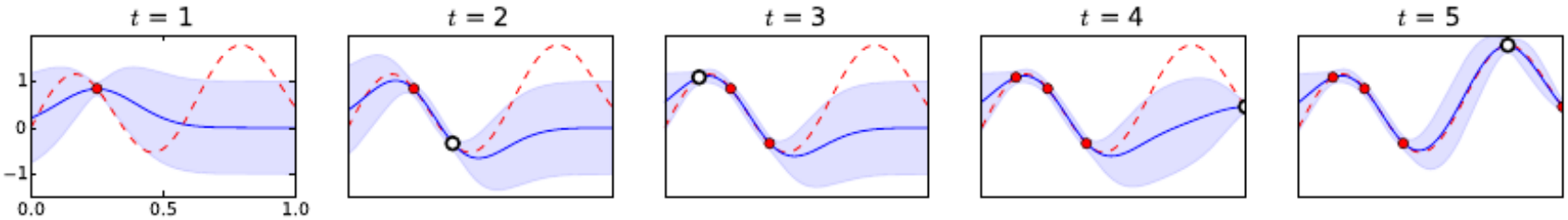
$$\text{argmax}_{x \in \mathcal{X}} f(x)$$

- **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.

- Optimal x_{n+1} through an **acquisition function**.

BO - upper confidence bound rule

- **UCB**: place x_{n+1} where **upper confidence bound is largest**.
- Strikes a balance between
  - **exploitation** (going for an higher function value)
  - **exploration** (x_{n+1} where uncertainty about f is large.)



Acquisition functions

- **Probability of Improvement (PI)**

$$a_{\text{PI}}(\boldsymbol{x}) \equiv \Pr\left(f(\boldsymbol{x}) > f_{\text{best}}\right) = 1 - \Phi\left(\frac{f_{\text{best}} - \hat{m}\left(\boldsymbol{x}; \mathcal{D}_n\right)}{s\left(\boldsymbol{x}; \mathcal{D}_n\right)}\right)$$

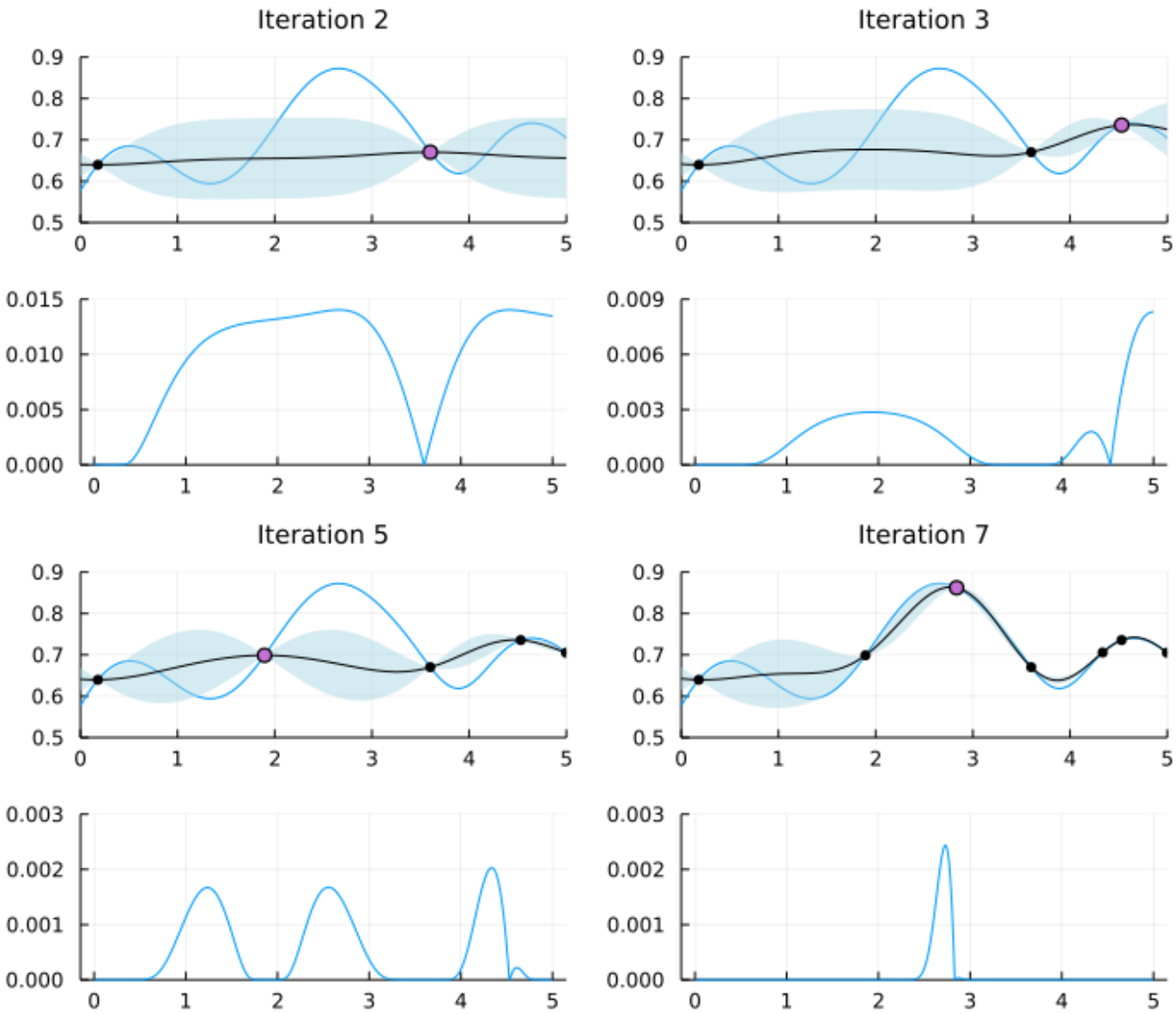
  - $\mathcal{D}_n = \{f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_n)\}$ are past evaluations
  - f_{best} is the smallest function value so far
  - $\hat{m}\left(\boldsymbol{x}; \mathcal{D}_n\right)$ is posterior mean of $f(\boldsymbol{x})$
  - $s\left(\boldsymbol{x}; \mathcal{D}_n\right)$ is posterior standard deviation of $f(\boldsymbol{x})$.

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

- **Expected Improvement per Second** - takes a **known function 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** $f(\boldsymbol{\theta}) \equiv \ln p(\boldsymbol{Y}_{1:T}|\boldsymbol{\theta})$ often estimated by **sampling:**
  - 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.
- **MCMC efficiency** and therefore $\mathbb{V}(\hat{f}(\boldsymbol{\theta}))$ **varies over** $\boldsymbol{\theta}$**-space**, particularly when $\boldsymbol{\theta}$ contains prior/regularization hyperparameters.
- **Stopping early** when probability of improvement (PI) is low.

Bayesian Optimization with Optimized Precision

- BOOP:
  - **Early stopping of evaluation** when **PI** $< \alpha$.
  - G **random** - we don't know G until we visit $\boldsymbol{\theta}$.
  - **EI per second**, but with G **predicted** for every $\boldsymbol{\theta}$.
- 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, e.g. the hyperparameter values themselves, or $\hat{m}(\boldsymbol{\theta}) - f_{\max}$.

The BOOP algorithm

- **BOOP acquisition function** from baseline $a(\boldsymbol{x})$ (e.g. EI):

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

a) Fit the heteroscedastic GP for f based on past evaluations

$$\hat{f}(\mathbf{x}_{1:(j-1)}) = f(\mathbf{x}_{1:(j-1)}) + \epsilon, \quad \epsilon \sim N(0, \Sigma_{1:(j-1)})$$
$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')),$$

where $\Sigma_{1:(j-1)} \equiv \text{Diag}(\sigma^2(G_1), \ldots, \sigma^2(G_{j-1}))$.

b) Fit the GP for $\log G$ based on past evaluations

$$\log G_{1:(j-1)} = h(\mathbf{z}_{1:(j-1)}) + \varepsilon, \quad \varepsilon \sim N(0, \psi^2 \mathbf{I})$$
$$h(\mathbf{z}) \sim \mathcal{GP}(m_G(\mathbf{z}), k_G(\mathbf{z}, \mathbf{z}')),$$

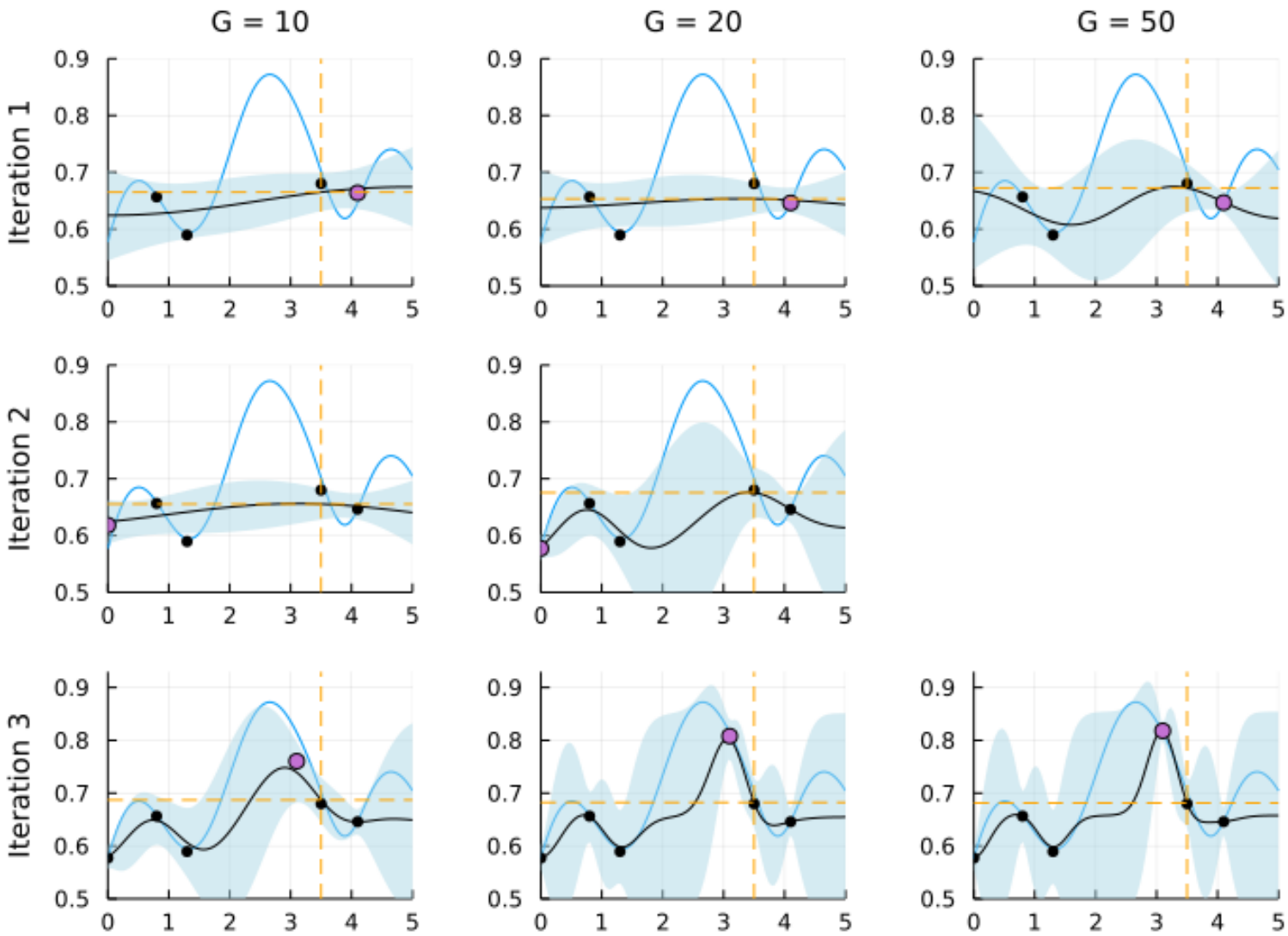
where the elements of $\mathbf{z}$ are functions of $\mathbf{x}$. Return the point prediction $\hat{G}_\alpha(\mathbf{x})$.

c) Maximize $\tilde{a}_\alpha(\mathbf{x}) = a(\mathbf{x})/\hat{G}_\alpha(\mathbf{x})$ to select the next point, $\mathbf{x}_j$.

d) Compute $\hat{f}(\mathbf{x}_j)$ and $\sigma^2(G_j)$ by early stopping at thresholding probability α.

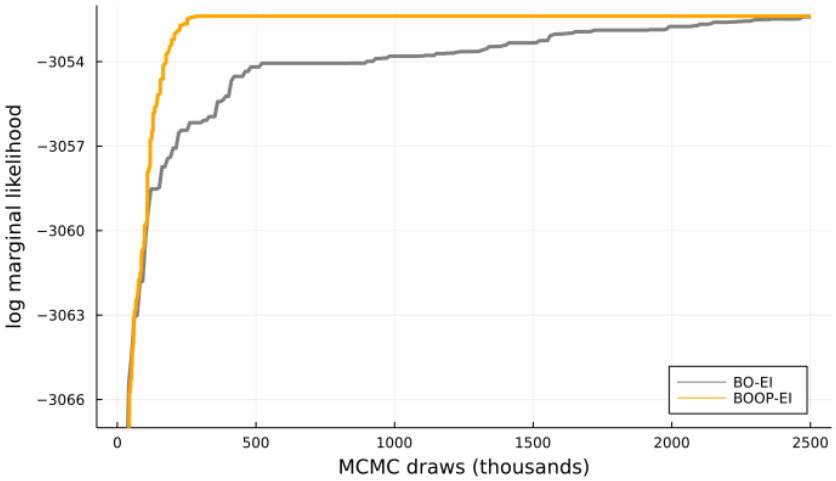
e) Update the datasets in a) with $(\mathbf{x}_j, \hat{f}(\mathbf{x}_j), \sigma^2(G_i))$ and in b) with $(\mathbf{z}_j, \log G_j)$.

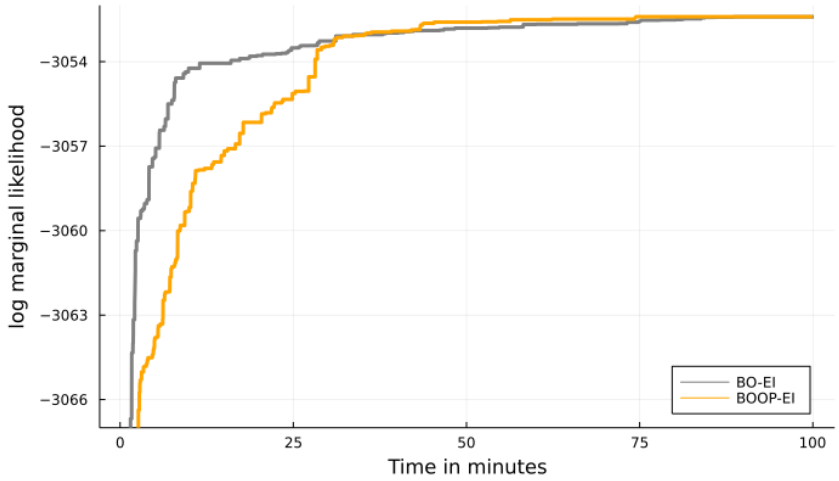
BOOP - illustration



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

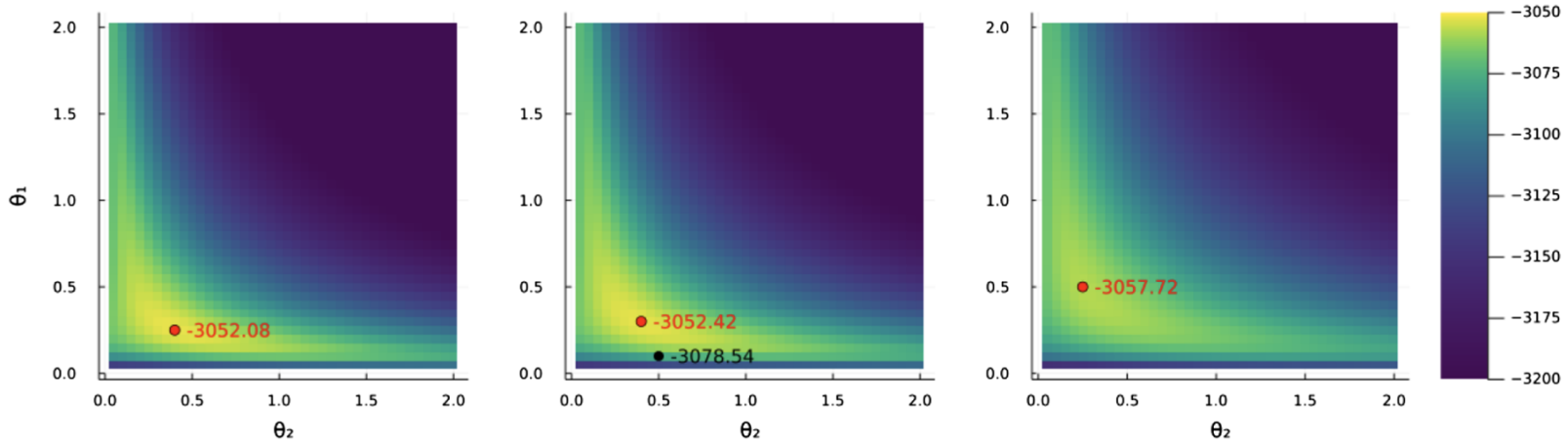


FIGURE 9 Log marginal likelihood surfaces over a fine grid of (θ_1, θ_2) values. The hyperparameter values for the lag decay are (a) $\theta_3 = 0.76$, (b) $\theta_3 = 1$, and (c) $\theta_3 = 2$ (left to right). The red dot denotes the maximum log marginal likelihood value for the given θ_3, and the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

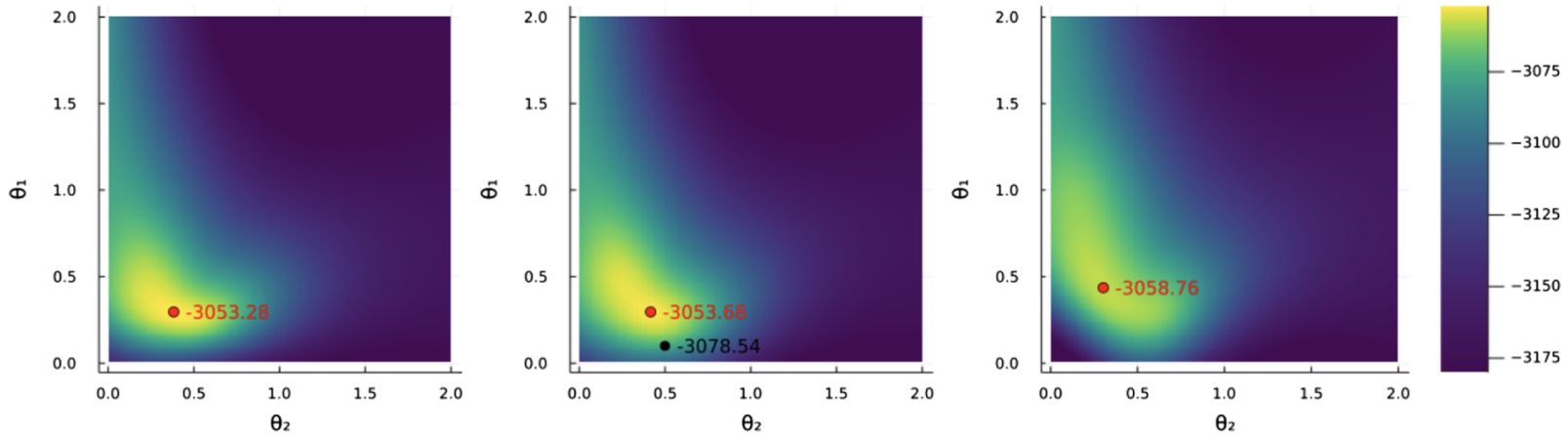


FIGURE 10 GP predictions of the hyperparameter surfaces in Figure 9 based on 250 evaluations for one BOOP-EI run. The hyperparameter for the lag decay is $\theta_3 = 0.76, 1$, and 2 (left to right). Red dot indicates the highest predicted value in the subplot, and the black dot, in the middle plot, shows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

22-variable steady-state BVAR

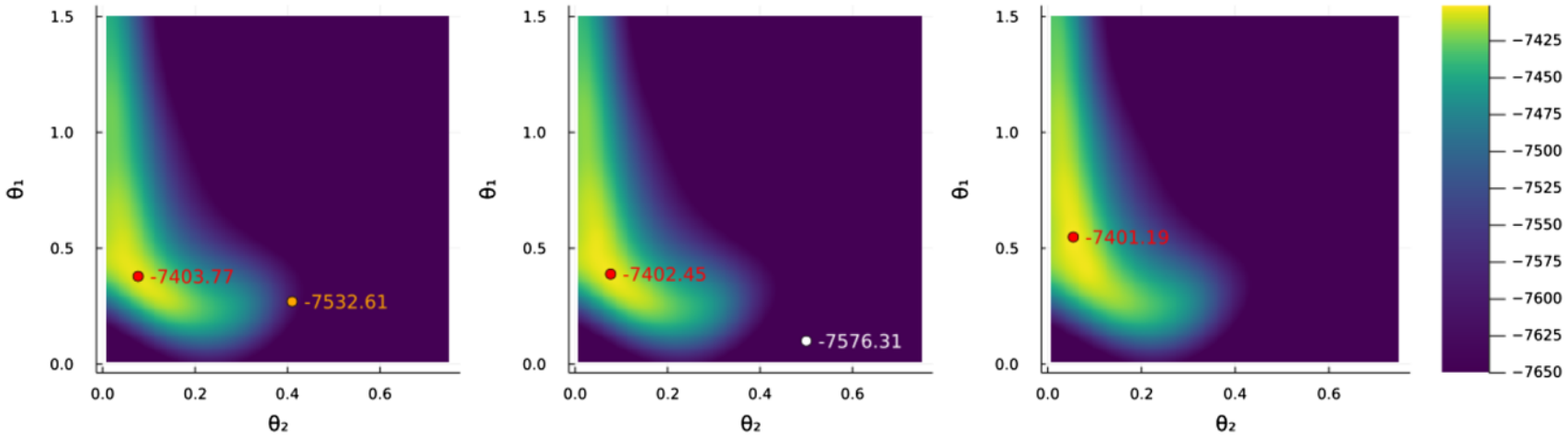


FIGURE 11 GP predictions of the hyperparameter surfaces for the large BVAR based on 250 iterations for a BOOP-EI run. The hyperparameter for the lag decay is $\theta_3 = 0.76$ (left graph, optimal in medium-sized BVAR), $\theta_3 = 1$ (middle graph, standard value), and $\theta_3 = 1.51$ (right, optimal for BOOP-EI). Red dot indicates the highest predicted value in a subplot. The orange dot in the leftmost plot shows the hyperparameters obtained from BOOP in the medium-sized BVAR, and the white dot in the middle plot shows the standard values.

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

- Time-varying parameter stochastic volatility BVAR:

$$A_0 y_t = c_t + \sum_{k=1}^{K} A_{k,t} y_{t-k} + \varepsilon_t, \quad \varepsilon_t \overset{\text{iid}}{\sim} N(0, \Sigma_t)$$

- Random walk evolution of $A_{k,t}$ and log variances.
- Three hyperparameters: prior mean of innovation variances (c_t, A_t and Σ_t).
- **Marginal likelihood** is 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

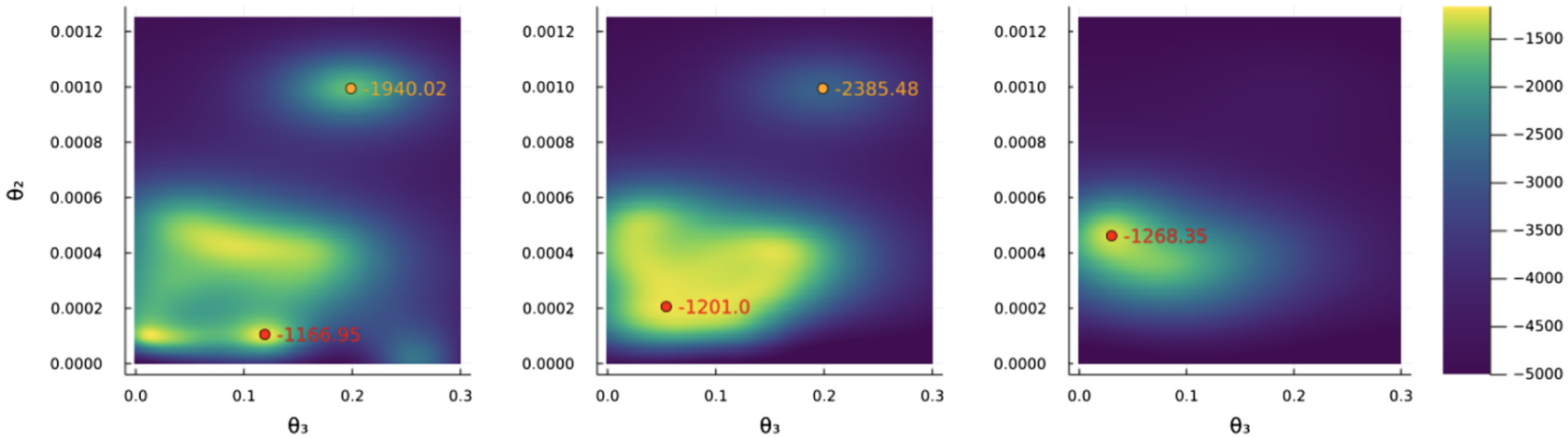


FIGURE 12 Predicted log marginal likelihood over the hyperparameters for stochastic volatility and the VAR dynamics for $\theta_1 = 0.0086$ (left), 0.05 (middle), and 0.1 (right). The mode in each plot is marked out by a red point. A distant local optimum is also marked out by an orange point.

Variational inference

- **Posterior approximation**. **Optimization** instead of sampling.
- **Mean field Variational Inference** (**MFVI**): approx posterior $p(\boldsymbol{\beta}|\boldsymbol{y}_{1:T})$ with factorized (independence) distribution $q(\boldsymbol{\beta})$

$$q(\boldsymbol{\beta}) = \prod_{i=1}^{p} q_i(\boldsymbol{\beta}_i)$$

- **Minimize Kullback-Leibler divergence** between p and q

$$KL(q, p) = E_q \left[\ln \frac{q(\boldsymbol{\beta})}{p(\boldsymbol{\beta}|\boldsymbol{y})} \right]$$

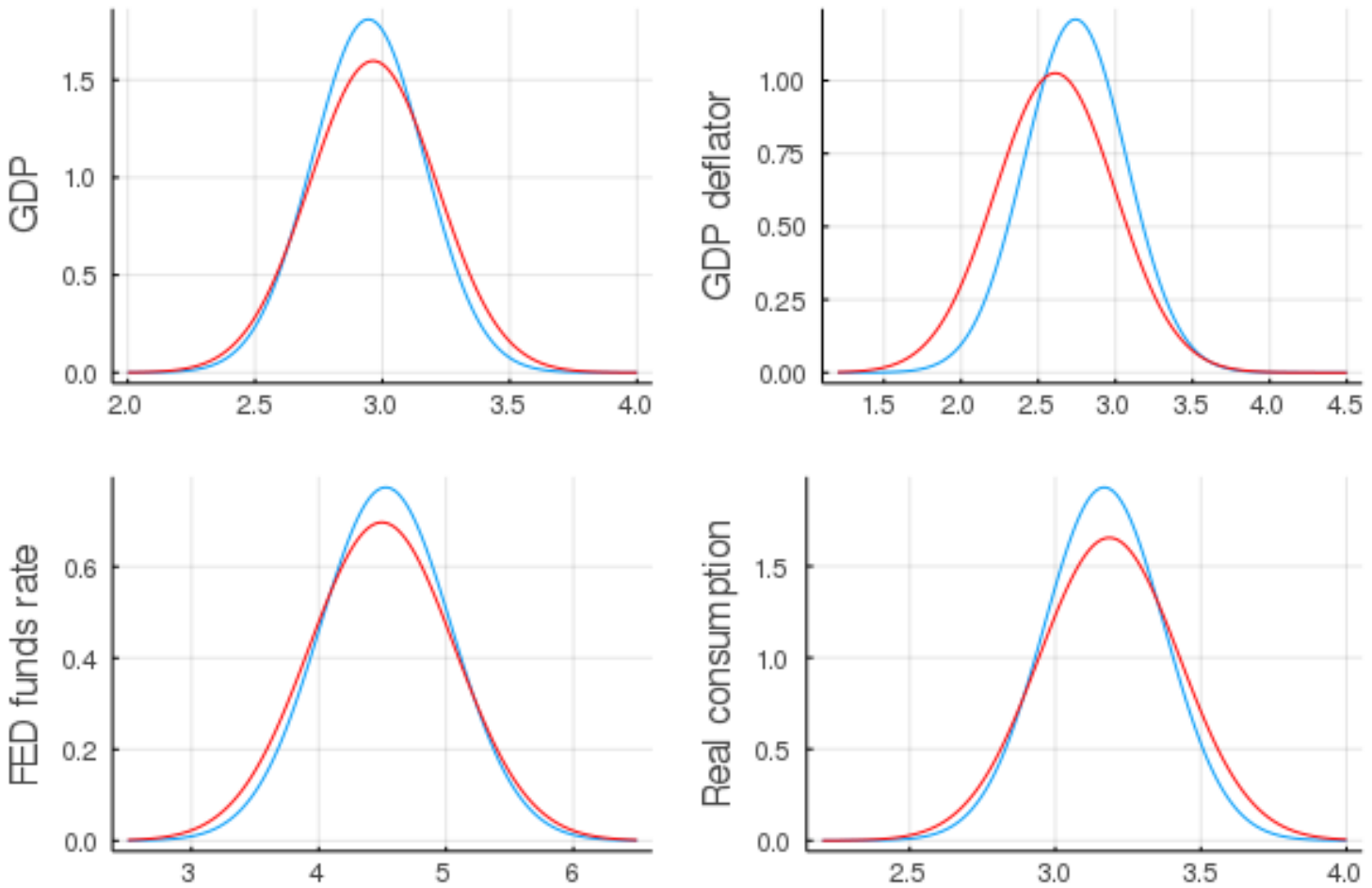
- Iterative updates to find minimum.
- **Structured MFVI** by blocking parameters

$$q(\boldsymbol{\beta}) = q_1(\boldsymbol{\beta}_1) q_2(\boldsymbol{\beta}_2)$$

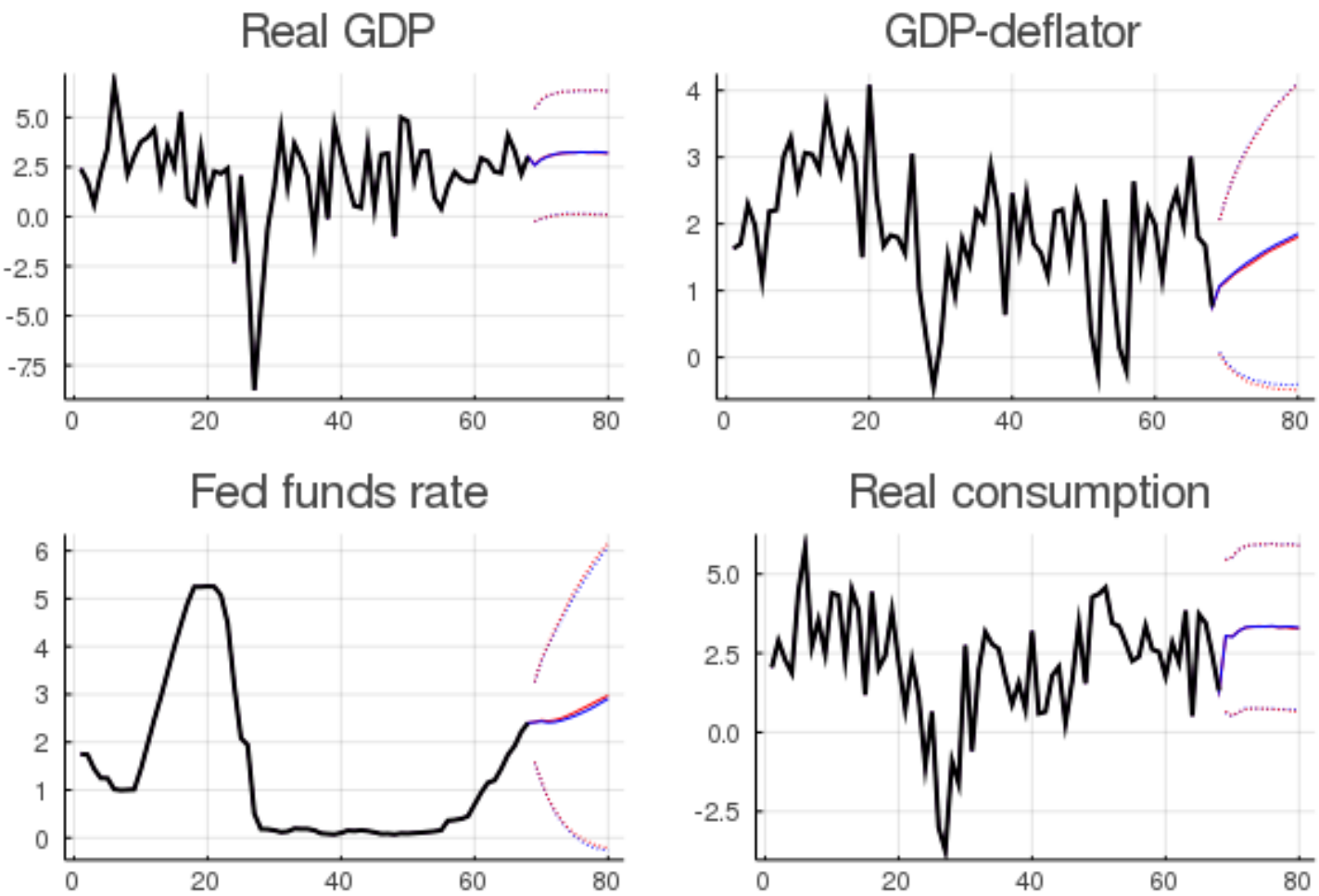
- Steady-state BVAR (closed-form updates):

$$q(\boldsymbol{A}, \boldsymbol{\Sigma}, \boldsymbol{\mu}) = q_A(\boldsymbol{A}) q_\Sigma(\boldsymbol{\Sigma}) q_\mu(\boldsymbol{\mu})$$

VI reasonably accurace for steady states



VI is very accurate for predictive distributions



VI for computing log predictive scores

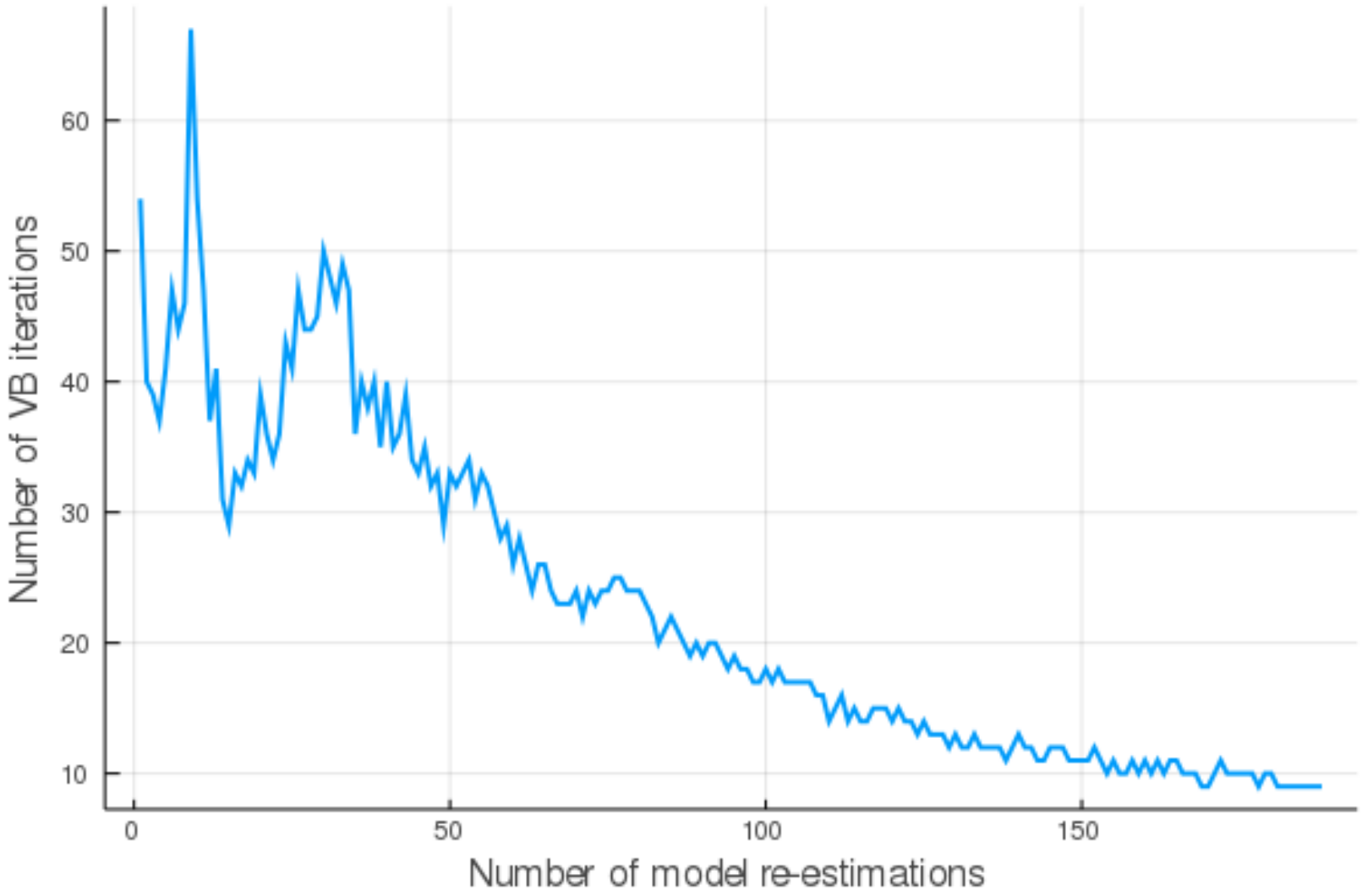
- Marginal likelihood can be sensitive to prior and overconfident (Oelrich et al, 2020).
- The **log predictive score** is a popular alternative

$$\text{LPS}_{t_\star} = \prod_{t=t_\star+1}^{T} p(y_t|y_{1:t-1})$$

$$p(y_t|y_{1:t-1}) = \int p(y_t|\boldsymbol{\beta}, y_{1:t-1}) p(\boldsymbol{\beta}|y_{1:t-1}) d\boldsymbol{\beta}$$

- Can also be used for **hyperparameter optimization**.
- Nott el al. (2012, JCGS): **VI benefits from warm start**: use solution from $t-1$ as initial value when optimizing $q(\boldsymbol{\beta}|y_{1:t}) \approx p(\boldsymbol{\beta}|y_{1:t})$.

Variational Inference benefits from warm starts



VI faster and scales better than MCMC

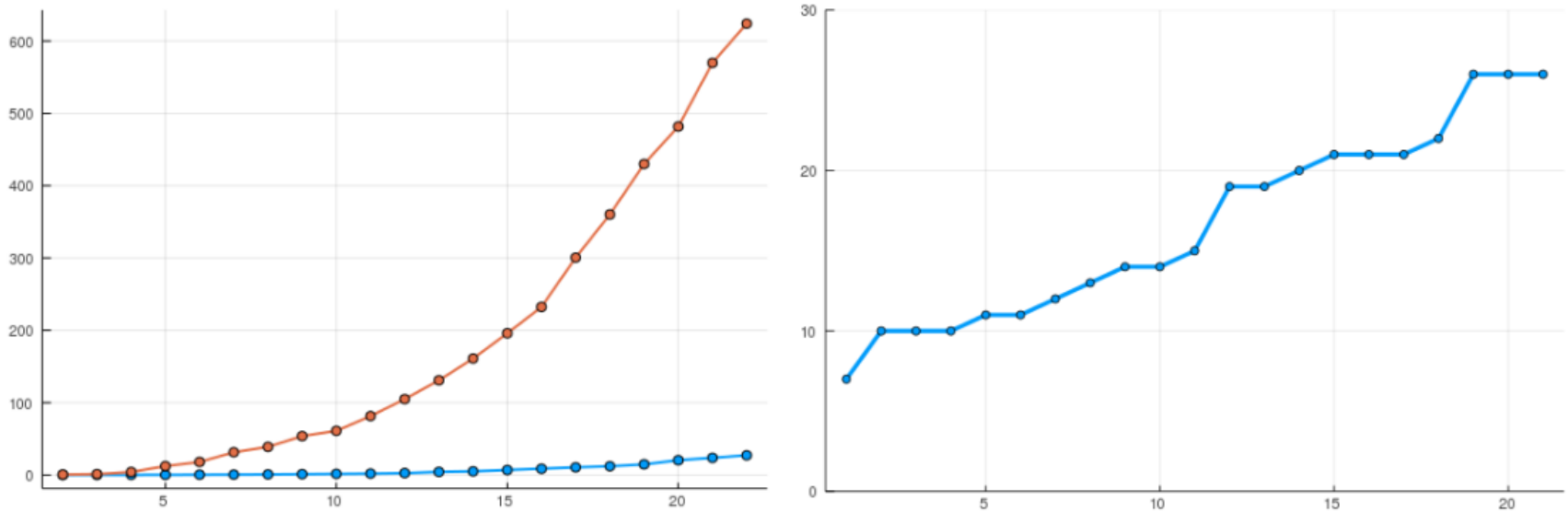
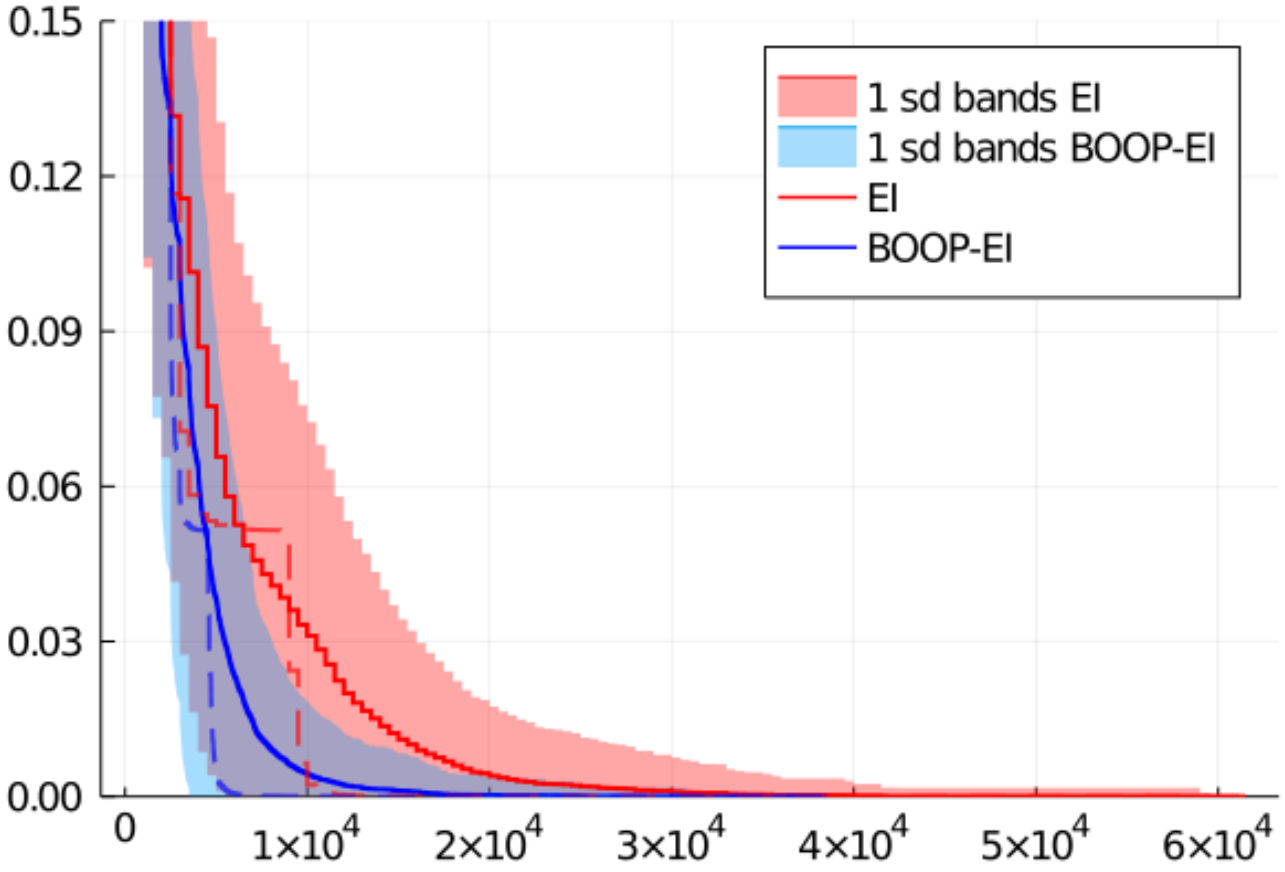


Figure 13: Computing times in seconds for the LPS with different number of time series in the VAR-system (left). The number of VI iterations until convergence as a function of the number of time series (right).

Conclusions

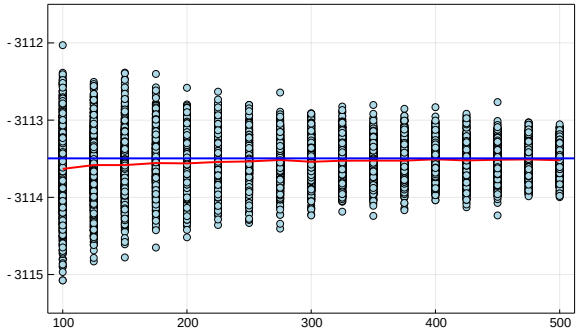
- **Bayesian optimization** is an attractive method for **costly**, **noisy**, **low-dimensional functions**.
- Hyperparameter optimization using **marginal likelihood estimated from MC sampling**.
- We extend BO to exploit that **the user controls the precision of the evaluations** via the number of samples.
- Succesful applications to steady-state BVARs and TVP-SV BVARs.
- **Future work**: develop "automatic" priors for hyperparameters and methods for exploring their roles and effects in models.

BOOP in action - repeated runs



Unbiased estimates

- BOOP **assumes (approx) unbiasedness** $\mathbb{E}\hat{f}(\boldsymbol{\theta}_i) = f(\boldsymbol{\theta}_i)$.
- **Sampling distribution of Chib's estimator** for SSBVAR.



- Unbiasedness depends on the Sampler-Estimator combination.
- Log marginal likelihood estimates in large-scale DSGE model: