

# Bayesian Hyperparameter Learning

**Mattias Villani**

Department of Statistics  
Stockholm University



[mattiasvillani.com](http://mattiasvillani.com)



@matvil



@matvil



mattiasvillani

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

# Hyperparameter inference - stats

- Parameter/Hyperparameter distinction:
  - ▶ **Parameters**,  $\beta$ , typically high-dim.
  - ▶ **Hyperparameters**  $\theta$ , typically low-dim.

## ■ L2-regularized linear regression

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

$$\beta_j | \sigma^2 \stackrel{\text{iid}}{\sim} N(\mathbf{0}, \tau^2 \sigma^2)$$

## ■ Horseshoe-regularized linear regression

$$\beta_j | \sigma^2 \stackrel{\text{iid}}{\sim} N(\mathbf{0}, \tau^2 \lambda_j^2 \sigma^2)$$

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

## ■ State-space models

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

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

# Hyperparameter inference - machine learning

## ■ Deep neural networks

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

## ■ Gaussian process regression/classification/spatial

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

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

# Hyperparameter inference - econometrics

- DSGE models in econometrics
  - ▶  $\beta$  are the persistence and variance of shocks etc
  - ▶  $\theta$  are parameters in the steady state.
- Bayesian **vector autoregressive models (VAR)** models

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

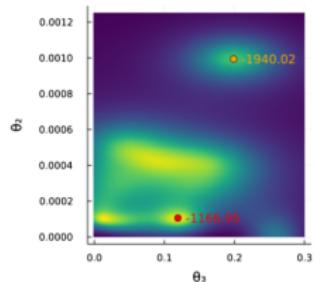
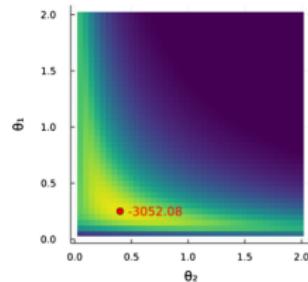
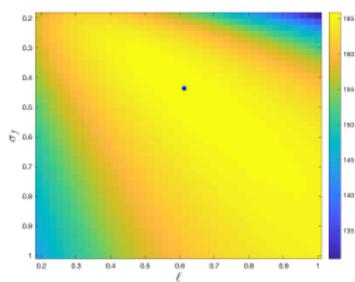
- ▶  $\beta = (\boldsymbol{\mu}, \mathbf{A}_1, \dots, \mathbf{A}_K, \boldsymbol{\Sigma})$
- ▶  $\theta = (\lambda_1, \lambda_2, \lambda_3)$  determine the prior standard deviation:

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

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

# Hyperparameter optimization

- Practitioners prefer to fix  $\theta$  “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**

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

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

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

# Hyperparameter optimization is tricky

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

# Gaussian processes regression

## ■ Gaussian process regression

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

## ■ Gaussian process prior over the space of functions

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$$

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

$$k(\mathbf{x}, \mathbf{x}') \equiv \text{Cov}(f(\mathbf{x}), f(\mathbf{x}')).$$

## ■ Squared exponential covariance function

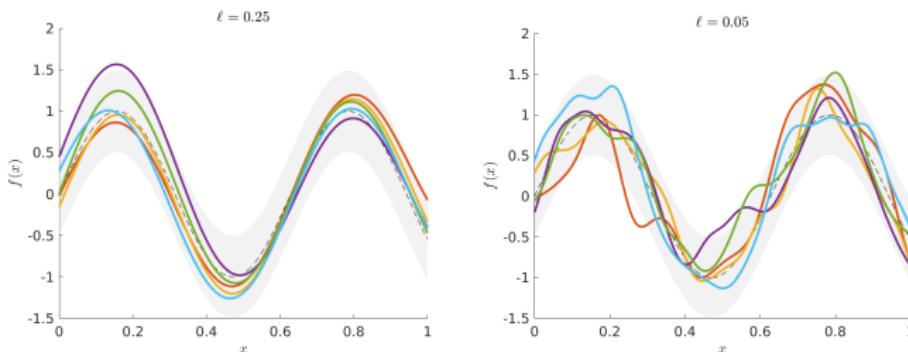
$$k(\mathbf{x}, \mathbf{x}') \equiv \text{Cov}(f(\mathbf{x}), f(\mathbf{x}')) = \sigma_f^2 \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\ell^2}\right)$$

# Nearby inputs $\mathbf{x}$ have correlated $f(\mathbf{x})$

- Squared exponential covariance function

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

- Correlation decay with  $\|\mathbf{x} - \mathbf{x}'\|$  depends on length scale  $\ell$ .



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

# Gaussian processes regression posterior

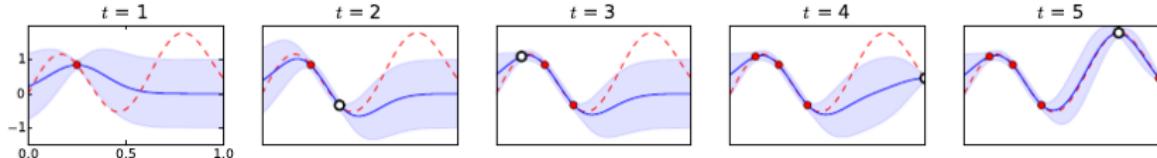
- Posterior of  $f(\cdot)$  at new test point  $\mathbf{x}_*$  is also Gaussian

$$\mathbf{f}_* | \mathbf{X}, \mathbf{y}, \mathbf{x}_* \sim N(\bar{\mathbf{f}}_*, \text{cov}(\mathbf{f}_*))$$

where

$$\begin{aligned}\bar{\mathbf{f}}_* &= k(\mathbf{x}_*, \mathbf{x}) [K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 I]^{-1} \mathbf{y} \\ \text{cov}(\mathbf{f}_*) &= k(\mathbf{x}_*, \mathbf{x}_*) - k(\mathbf{x}_*, \mathbf{x}) [K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 I]^{-1} k(\mathbf{x}, \mathbf{x}_*)\end{aligned}$$

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



# Bayesian optimization

- Aim: **maximization of expensive function**

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

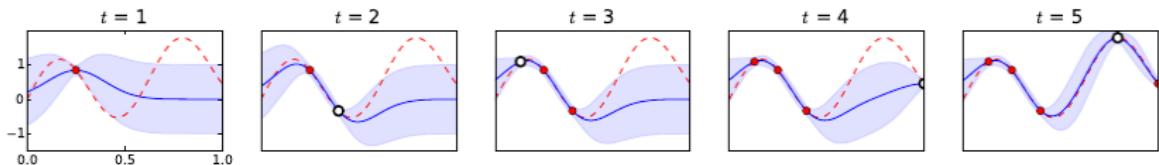
- **Bayesian optimization:**

- ▶ Assume  $f \sim \mathcal{GP}$
- ▶ Evaluate  $f$  at  $x_1, x_2, \dots, x_n$ .
- ▶ Update to posterior distribution  $f|x_1, \dots, x_n \sim \mathcal{GP}$ .
- ▶ 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 a higher function value)
  - ▶ **exploration** ( $x_{n+1}$  where uncertainty about  $f$  is large.)



# Acquisition functions

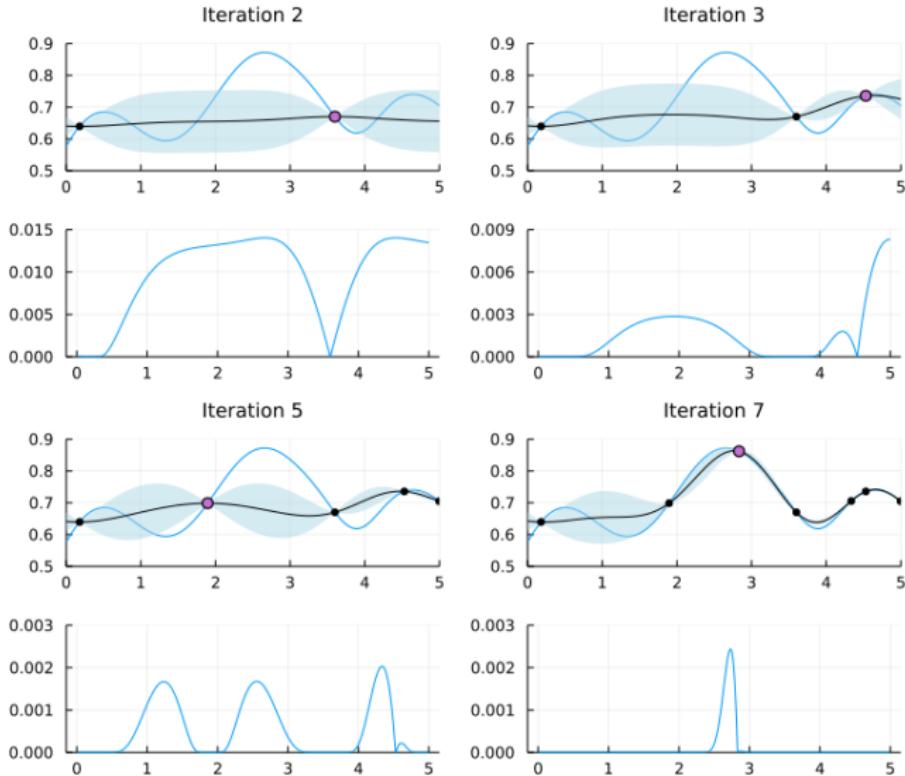
## ■ Probability of Improvement (PI)

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

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

- 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  $f(\theta) \equiv \ln p(\mathbf{Y}_{1:T} | \theta)$  often estimated by sampling:
  - ▶ Chib (Gibbs) and Chib-Jeliazkov (MH)
  - ▶ Importance sampling,
  - ▶ Particle filters
- Noisy evaluations  $\hat{f}(\theta)$ .
- Precision of  $\hat{f}(\theta)$  controlled via number of samples  $G$ .
- MCMC efficiency and therefore  $\mathbb{V}(\hat{f}(\theta))$  varies over  $\theta$ -space, particularly when  $\theta$  contains prior/regularization hyperparameters.
- Stopping early when probability of improvement (PI) is low.

# Bayesian Optimization with Optimized Precision

## ■ BOOP:

- ▶ Early stopping of evaluation when  $\text{PI} < \alpha$ .
- ▶  $G$  random - we don't know  $G$  until we visit  $\theta$ .
- ▶ EI per second, but with  $G$  predicted for every  $\theta$ .

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

$$\hat{f}(\theta_i) = f(\theta_i) + \epsilon_i, \quad \epsilon_i \stackrel{\text{iid}}{\sim} N(0, \sigma^2(G_i))$$

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

$$\ln G_i = h(\mathbf{z}_i) + \varepsilon_i \quad \varepsilon_i \stackrel{\text{iid}}{\sim} N(0, \psi^2),$$

where  $\mathbf{z}$  are variables with predictive power for  $G$ , e.g. the hyperparameter values themselves, or  $\hat{m}(\theta) - f_{\max}$ .

# The BOOP algorithm

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

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

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

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

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

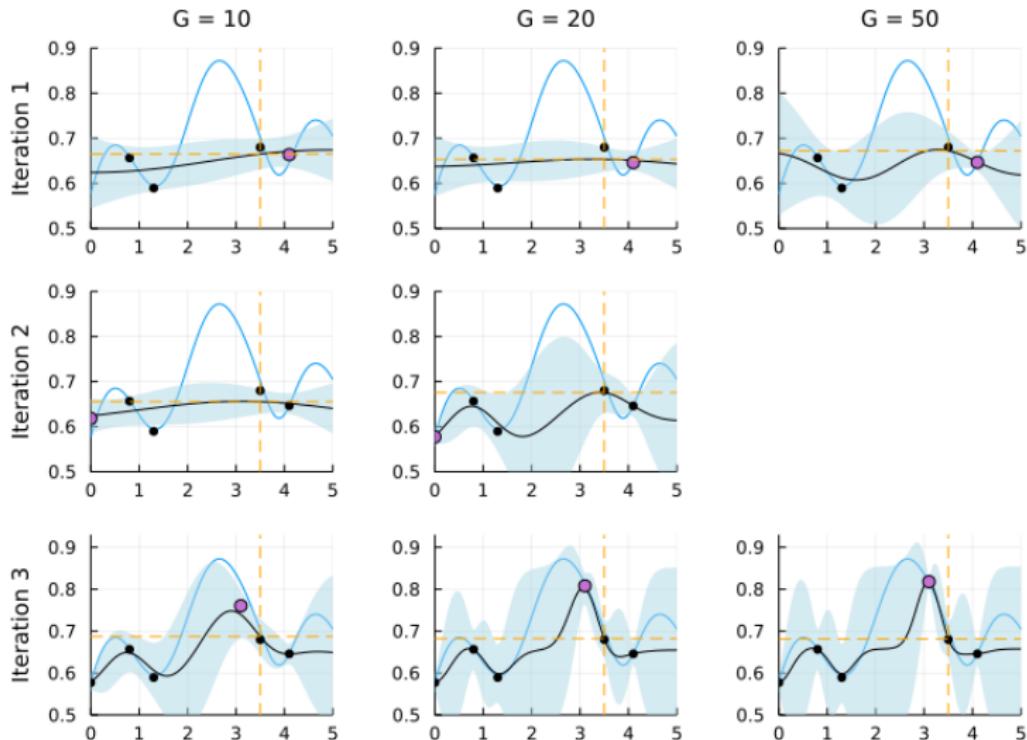
- b) Fit the GP for  $\log G$  based on past evaluations

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

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

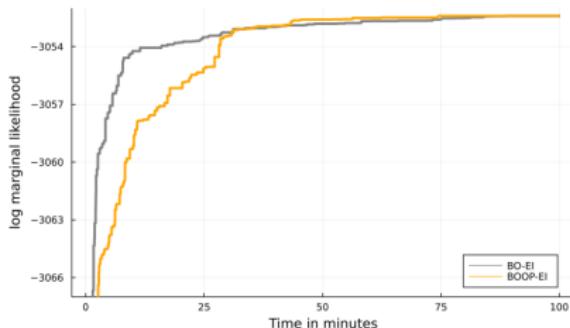
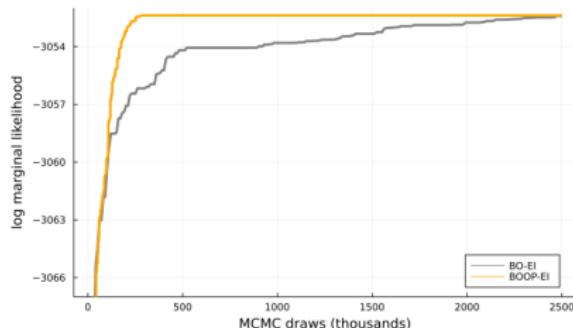
- c) Maximize  $\tilde{a}_\alpha(\mathbf{x}) = a(\mathbf{x})/\hat{G}_\alpha(\mathbf{x})$  to select the next point,  $\mathbf{x}_j$ .
- d) Compute  $\hat{f}(\mathbf{x}_j)$  and  $\sigma^2(G_j)$  by early stopping at thresholding probability  $\alpha$ .
- e) Update the datasets in a) with  $(\mathbf{x}_j, \hat{f}(\mathbf{x}_j), \sigma^2(G_j))$  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  $\theta = (\lambda_1, \lambda_2, \lambda_3)$ .



# 7-variable BVAR - true ML surface vs predicted

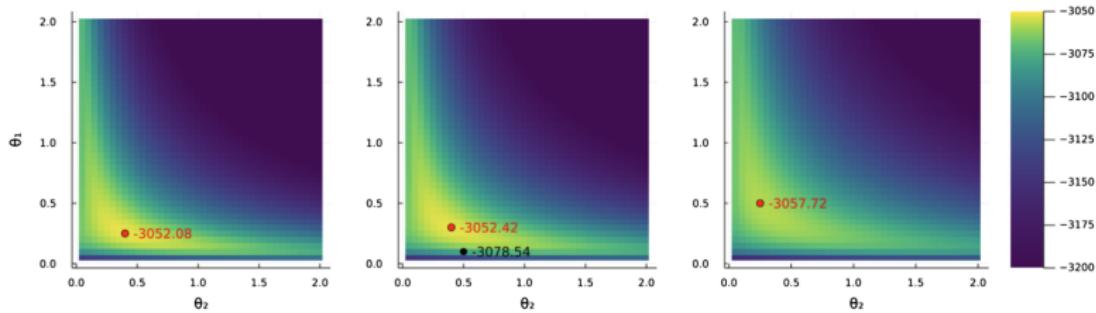


FIGURE 9 Log marginal likelihood surfaces over a fine grid of  $(\theta_1, \theta_2)$  values. The hyperparameter values for the lag decay are (a)  $\theta_3 = 0.76$ , (b)  $\theta_3 = 1$ , and (c)  $\theta_3 = 2$  (left to right). The red dot denotes the maximum log marginal likelihood value for the given  $\theta_3$ , and the black dot, in the middle plot, shows the standard values.

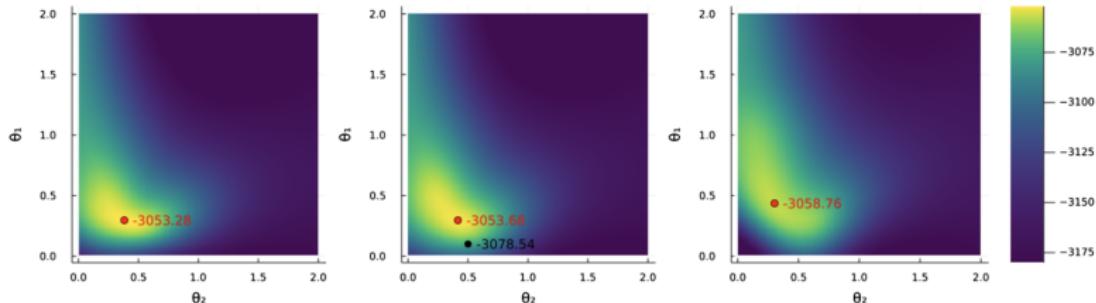


FIGURE 10 GP predictions of the hyperparameter surfaces in Figure 9 based on 250 evaluations for one BOOP-EI run. The hyperparameter for the lag decay is  $\theta_3 = 0.76, 1$ , and  $2$  (left to right). Red dot indicates the highest predicted value in the subplot, and the black dot, in the middle plot,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

## 22-variable steady-state BVAR

	<b>Standard</b>	<b>BO-EI</b>	<b>BOOP-EI</b>	<b>Medium BVAR</b>
Log ML	-7576.31	-7402.50	-7401.09	-7532.61
Sd log ML	0.54	0.81	0.16	0.49
Gibbs iterations		$3.75 \times 10^6$	$1.8 \times 10^6$	
CPU time (h)		64.90	20.22	
$\theta_1$	0.1	0.47	0.56	0.27
$\theta_2$	0.5	0.06	0.05	0.41
$\theta_3$	1	1.46	1.51	0.76

## 22-variable steady-state BVAR

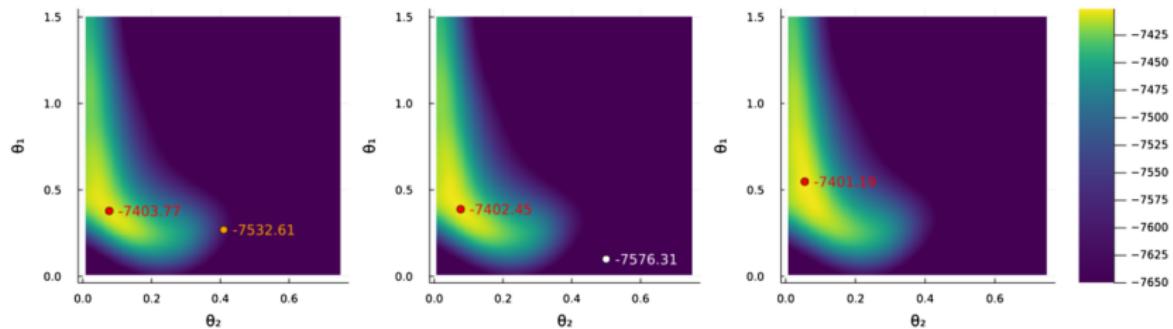


FIGURE 11 GP predictions of the hyperparameter surfaces for the large BVAR based on 250 iterations for a BOOP-EI run. The hyperparameter for the lag decay is  $\theta_3 = 0.76$  (left graph, optimal in medium-sized BVAR),  $\theta_3 = 1$  (middle graph, standard value), and  $\theta_3 = 1.51$  (right, 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:

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

- 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.

	<b>CE</b>	<b>BO1</b>	<b>BO2</b>	<b>BO3</b>	<b>BOOP1</b>	<b>BOOP2</b>	<b>BOOP3</b>
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

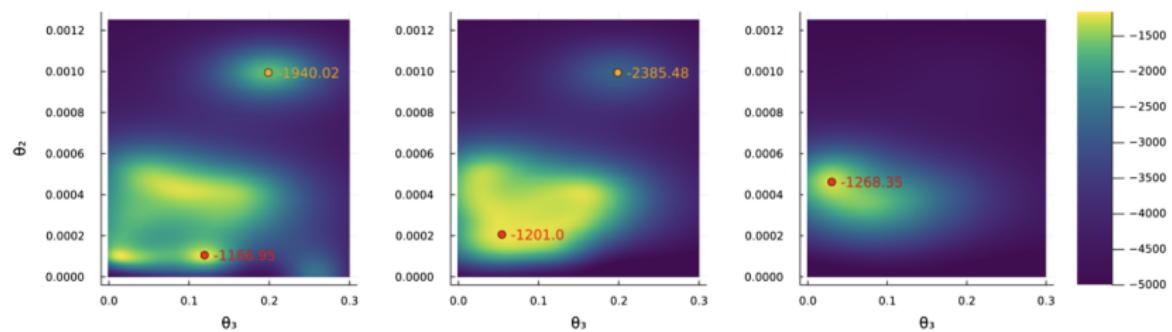


FIGURE 12 Predicted log marginal likelihood over the hyperparameters for stochastic volatility and the VAR dynamics for  $\theta_1 = 0.0086$  (left), 0.05 (middle), and 0.1 (right). The mode in each plot is marked out by a red point. A distant local optimum is also marked out by an orange point.

# Variational inference

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

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

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

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

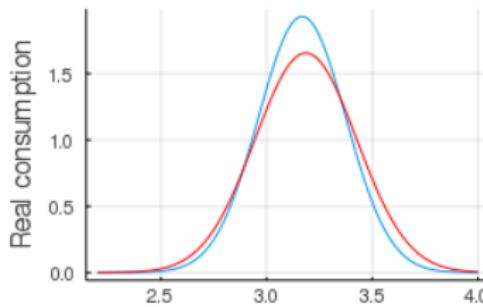
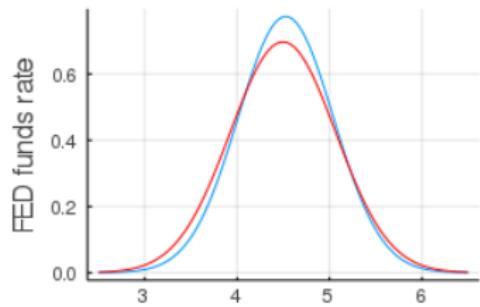
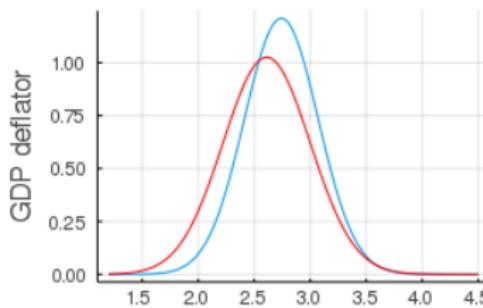
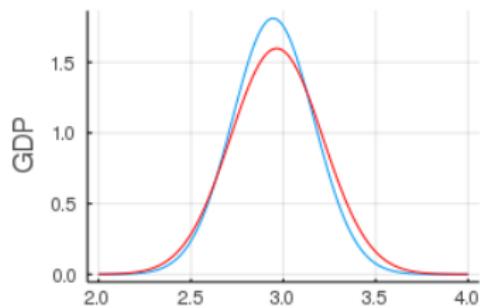
- Iterative updates to find minimum.
- Structured MFVI by blocking parameters

$$q(\beta) = q_1(\beta_1)q_2(\beta_2)$$

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

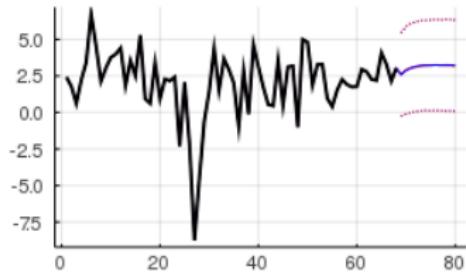
$$q(\mathbf{A}, \Sigma, \mu) = q_A(\mathbf{A})q_\Sigma(\Sigma)q_\mu(\mu)$$

# VI reasonably accurate for steady states

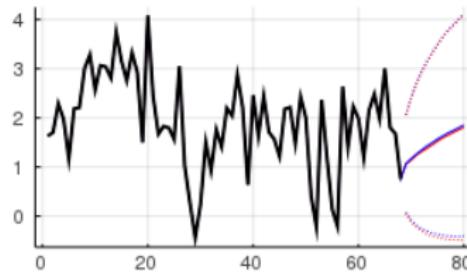


# VI is very accurate for predictive distributions

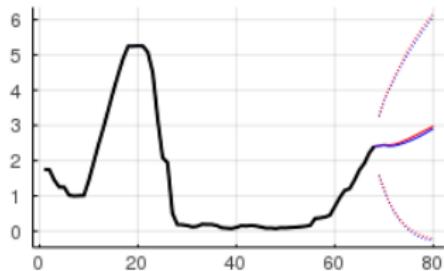
Real GDP



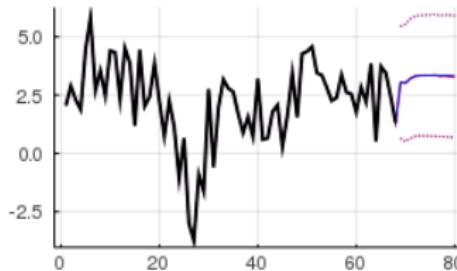
GDP-deflator



Fed funds rate



Real consumption



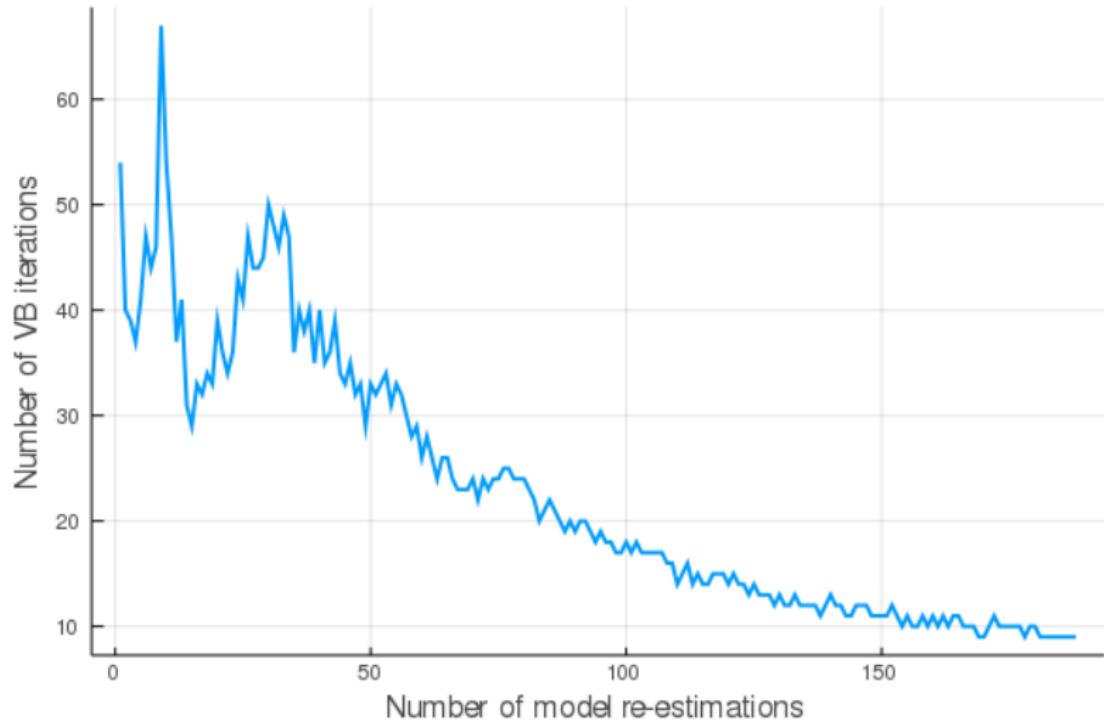
## VI for computing log predictive scores

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

$$\text{LPS}_{t_*} = \prod_{t=t_*+1}^T p(y_t | y_{1:t-1})$$
$$p(y_t | y_{1:t-1}) = \int p(y_t | \beta, y_{1:t-1}) p(\beta | y_{1:t-1}) d\beta$$

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

# Variational Inference benefits from warm starts



# VI faster and scales better than MCMC

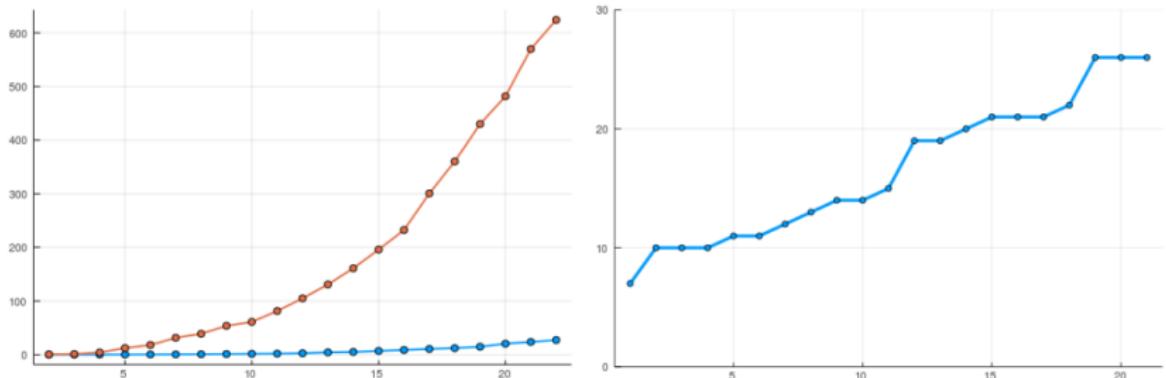
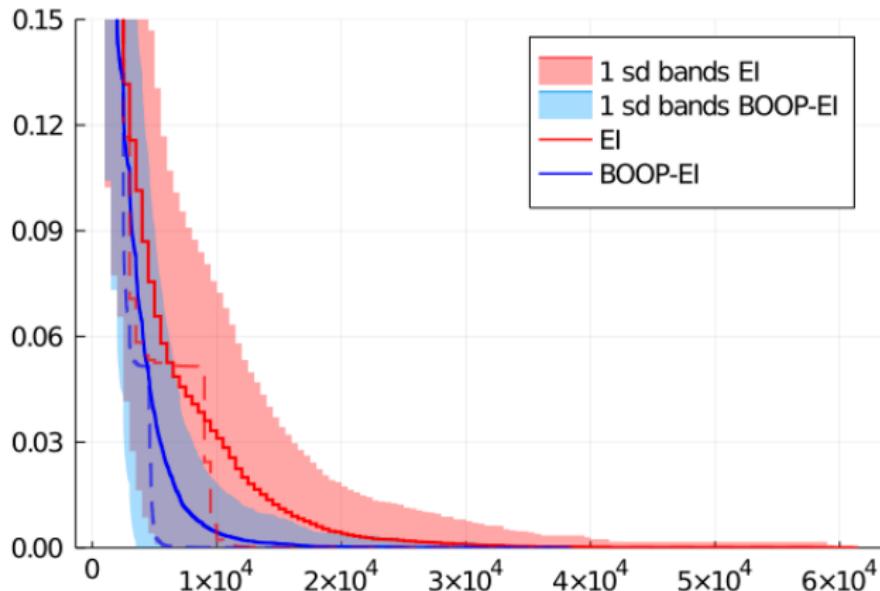


Figure 13: Computing times in seconds for the LPS with different number of time series in the VAR-system (left). The number of VI iterations until convergence as a function of the number of time series (right).

# Conclusions

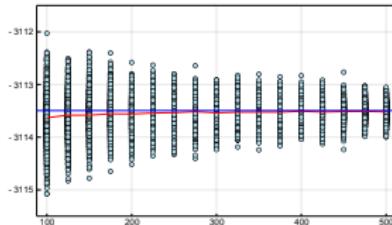
- Bayesian optimization is an attractive method for **costly, noisy, low-dimensional functions.**
- Hyperparameter optimization using **marginal likelihood estimated from MC sampling.**
- We extend BO to exploit that **the user controls the precision of the evaluations** via the number of samples.
- Successful 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  $\mathbb{E}\hat{f}(\theta_i) = f(\theta_i)$ .
- Sampling distribution of Chib's estimator for SSBVAR.



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

