Variational Bayesian Monte Carlo

Luigi Acerbi

Department of Basic Neuroscience
University of Geneva

Nov 26, 2018

1. Introduction and motivation
2. Background Tools
3. Variational Bayesian Monte Carlo
4. Experiments

1 Introduction and motivation

2 Background Tools

3 Variational Bayesian Monte Carlo

4 Experiments

Goal

Bayesian inference with expensive black-box statistical models

Goal

Bayesian inference with expensive black-box statistical models

- Likelihood: $p(\mathcal{D}|\boldsymbol{x})$ (data $\mathcal{D}$, parameters $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^D$)

Goal

Bayesian inference with expensive black-box statistical models

- Likelihood: $p(\mathcal{D}|\boldsymbol{x})$ (data $\mathcal{D}$, parameters $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^D$)
- No detailed information (e.g., no gradient)

Goal

Bayesian inference with expensive black-box statistical models

- Likelihood: $p(\mathcal{D}|\boldsymbol{x})$ (data $\mathcal{D}$, parameters $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^D$)
- No detailed information (e.g., no gradient)
- $\sim$ 500–1000 likelihood evaluations

Goal

Bayesian inference with expensive black-box statistical models

- Likelihood: $p(\mathcal{D}|\boldsymbol{x})$ (data $\mathcal{D}$, parameters $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^D$)
- No detailed information (e.g., no gradient)
- $\sim$ 500–1000 likelihood evaluations

Posterior: $p(\boldsymbol{x}|\mathcal{D}) = \frac{p(\mathcal{D}|\boldsymbol{x})p(\boldsymbol{x})}{p(\mathcal{D})}$ (in usable form)

Marginal likelihood: $p(\mathcal{D}) = \int p(\mathcal{D}|\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x}$

Goal

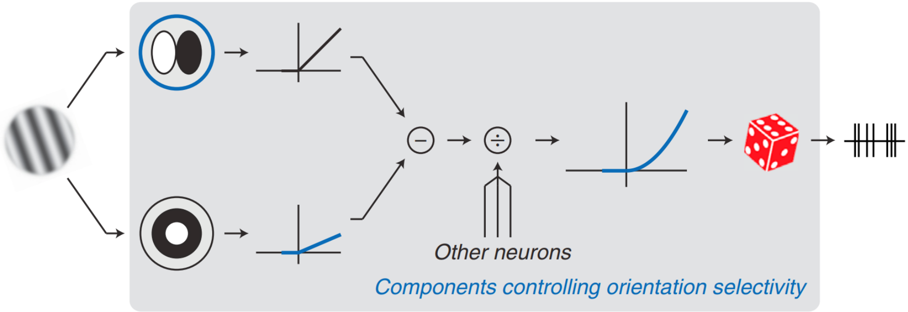
Bayesian inference with expensive black-box statistical models

- Likelihood: $p(\mathcal{D}|\boldsymbol{x})$ (data $\mathcal{D}$, parameters $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^D$)
- No detailed information (e.g., no gradient)
- $\sim$ 500–1000 likelihood evaluations

Posterior: $p(\boldsymbol{x}|\mathcal{D}) = \frac{p(\mathcal{D}|\boldsymbol{x})p(\boldsymbol{x})}{p(\mathcal{D})}$ (in usable form)
Marginal likelihood: $p(\mathcal{D}) = \int p(\mathcal{D}|\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x}$

(Why Bayesian inference?)

Example: LN-LN neuronal model



from Goris et al., *Neuron* (2015)

Problem

Bayesian inference with expensive black-box statistical models?

Problem

Bayesian inference with expensive black-box statistical models?

Standard approximate Bayesian inference methods

- Markov Chain Monte Carlo (MCMC)
- Variational inference (VI)

Problem

Bayesian inference with expensive black-box statistical models?

Standard approximate Bayesian inference methods

- Markov Chain Monte Carlo (MCMC)
- Variational inference (VI)

require

- many likelihood evaluations

Problem

Bayesian inference with expensive black-box statistical models?

Standard approximate Bayesian inference methods

- Markov Chain Monte Carlo (MCMC)
- Variational inference (VI)

require

- many likelihood evaluations
- knowledge of the model (e.g., gradients, detailed structure)

Sketch solution

Bayesian inference with expensive black-box statistical models?

Sketch solution

Bayesian inference with expensive black-box statistical models?

- Fit *surrogate model* to likelihood evaluations

Sketch solution

Bayesian inference with expensive black-box statistical models?

- Fit *surrogate model* to likelihood evaluations
- Perform *approximate inference* with surrogate model

Sketch solution

Bayesian inference with expensive black-box statistical models?

- Fit *surrogate model* to likelihood evaluations
- Perform *approximate inference* with surrogate model
- Use *active sampling* to smartly evaluate likelihood landscape

What do we need?

- An *approximate inference* framework
- A *surrogate model*
- A method to combine the two

What do we need?

- An *approximate inference* framework: variational inference
- A *surrogate model*: Gaussian processes
- A method to combine the two: Bayesian quadrature

1 Introduction and motivation

2 Background Tools

3 Variational Bayesian Monte Carlo

4 Experiments

Variational inference

- Approximate $p(\boldsymbol{x}|\mathcal{D})$ with $q_\phi(\boldsymbol{x})$

Variational inference

- Approximate $p(\boldsymbol{x}|\mathcal{D})$ with $q_\phi(\boldsymbol{x})$
- Minimize $\mathrm{KL}\left[q_\phi(\boldsymbol{x})||p(\boldsymbol{x}|\mathcal{D})\right] = \mathbb{E}_{q_\phi}\left[\log \frac{q_\phi(\boldsymbol{x})}{p(\boldsymbol{x}|\mathcal{D})}\right]$

Variational inference

- Approximate $p(\boldsymbol{x}|\mathcal{D})$ with $q_\phi(\boldsymbol{x})$
- Minimize $\text{KL}\left[q_\phi(\boldsymbol{x})||p(\boldsymbol{x}|\mathcal{D})\right] = \mathbb{E}_{q_\phi}\left[\log \frac{q_\phi(\boldsymbol{x})}{p(\boldsymbol{x}|\mathcal{D})}\right]$

$$\implies \text{Maximize ELBO}(\phi) = \underbrace{\mathbb{E}_{q_\phi}\left[\log p(\mathcal{D}|\boldsymbol{x})p(\boldsymbol{x})\right]}_{\text{expected log joint}} + \underbrace{\mathcal{H}[q_\phi(\boldsymbol{x})]}_{\text{entropy}}$$

Variational inference

- Approximate $p(\boldsymbol{x}|\mathcal{D})$ with $q_\phi(\boldsymbol{x})$
- Minimize $\text{KL}\left[q_\phi(\boldsymbol{x})||p(\boldsymbol{x}|\mathcal{D})\right] = \mathbb{E}_{q_\phi}\left[\log \frac{q_\phi(\boldsymbol{x})}{p(\boldsymbol{x}|\mathcal{D})}\right]$

$$\Longrightarrow \text{Maximize ELBO}(\phi) = \underbrace{\mathbb{E}_{q_\phi}\left[\log p(\mathcal{D}|\boldsymbol{x})p(\boldsymbol{x})\right]}_{\text{expected log joint}} + \underbrace{\mathcal{H}[q_\phi(\boldsymbol{x})]}_{\text{entropy}} \leq \log p(\mathcal{D})$$

Variational inference

- Approximate $p(\boldsymbol{x}|\mathcal{D})$ with $q_\phi(\boldsymbol{x})$
- Minimize $\mathrm{KL}\left[q_\phi(\boldsymbol{x})||p(\boldsymbol{x}|\mathcal{D})\right] = \mathbb{E}_{q_\phi}\left[\log \frac{q_\phi(\boldsymbol{x})}{p(\boldsymbol{x}|\mathcal{D})}\right]$

$$\Longrightarrow \text{Maximize } \mathrm{ELBO}(\phi) = \underbrace{\mathbb{E}_{q_\phi}\left[\log p(\mathcal{D}|\boldsymbol{x})p(\boldsymbol{x})\right]}_{\text{expected log joint}} + \underbrace{\mathcal{H}[q_\phi(\boldsymbol{x})]}_{\text{entropy}} \leq \log p(\mathcal{D})$$

Obtains

- An approximate posterior $q_\phi(\boldsymbol{x})$
- A lower bound to the log marginal likelihood, $\mathrm{ELBO}(\phi)$

Variational inference

- Approximate $p(\boldsymbol{x}|\mathcal{D})$ with $q_\phi(\boldsymbol{x})$
- Minimize $\mathrm{KL}\left[q_\phi(\boldsymbol{x})||p(\boldsymbol{x}|\mathcal{D})\right] = \mathbb{E}_{q_\phi}\left[\log \frac{q_\phi(\boldsymbol{x})}{p(\boldsymbol{x}|\mathcal{D})}\right]$

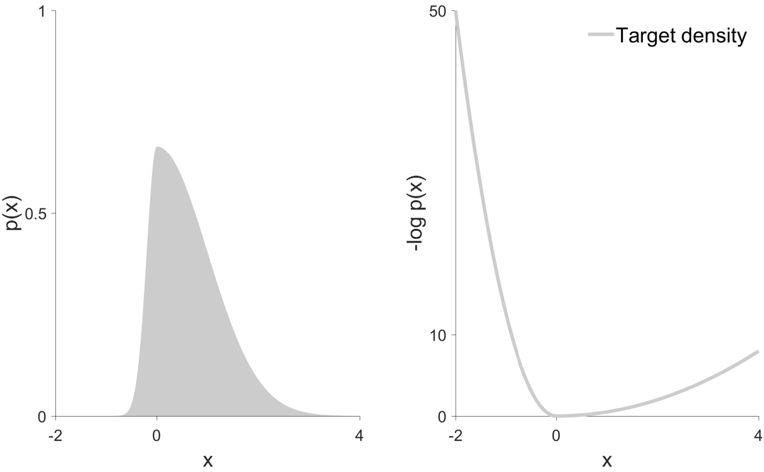
$$\Longrightarrow \text{Maximize } \mathrm{ELBO}(\phi) = \underbrace{\mathbb{E}_{q_\phi}\left[\log p(\mathcal{D}|\boldsymbol{x})p(\boldsymbol{x})\right]}_{\text{expected log joint}} + \underbrace{\mathcal{H}[q_\phi(\boldsymbol{x})]}_{\text{entropy}} \leq \log p(\mathcal{D})$$

Obtains

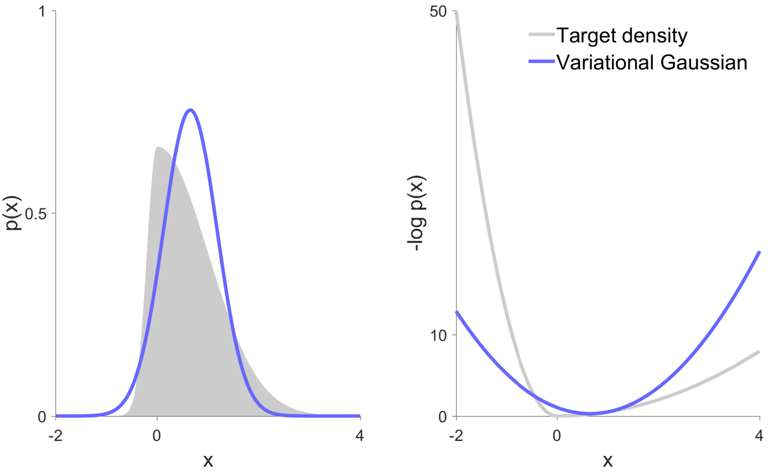
- An approximate posterior $q_\phi(\boldsymbol{x})$
- A lower bound to the log marginal likelihood, $\mathrm{ELBO}(\phi)$

VI casts Bayesian inference into optimization + integration

Variational inference: example

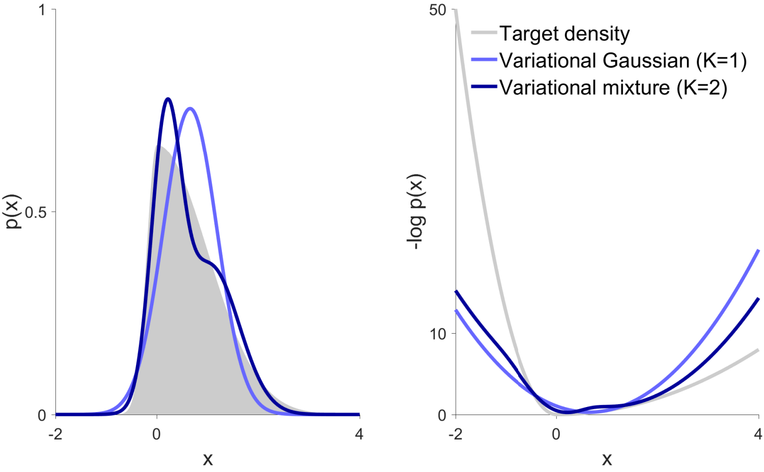


Variational inference: example



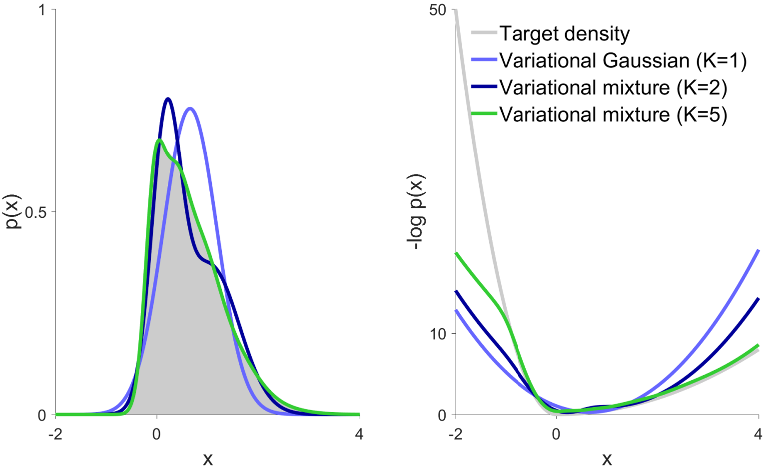
$$q_\phi(x) = \mathcal{N}\left(x, \mu, \sigma^2\right) \qquad \phi = \left(\mu, \sigma^2\right)$$

Variational inference: example



$$q_\phi(x) = \sum_{k=1}^{K} w_k \mathcal{N}\left(x, \mu_k, \sigma_k^2\right) \qquad \phi = \left(w_k, \mu_k, \sigma_k^2\right)_{k=1}^{K}$$

Variational inference: example



$$q_\phi(x) = \sum_{k=1}^{K} w_k \mathcal{N}\left(x, \mu_k, \sigma_k^2\right) \qquad \phi = \left(w_k, \mu_k, \sigma_k^2\right)_{k=1}^{K}$$

Gaussian Processes (GPs)

GPs used as *priors* over $f : \mathcal{X} \subseteq \mathbb{R}^D \to \mathbb{R}$

Gaussian Processes (GPs)

GPs used as *priors* over $f : \mathcal{X} \subseteq \mathbb{R}^D \to \mathbb{R}$

- mean function $m : \mathcal{X} \to \mathbb{R}$
  - zero, constant, polynomial...

Gaussian Processes (GPs)

GPs used as *priors* over $f : \mathcal{X} \subseteq \mathbb{R}^D \to \mathbb{R}$

- mean function $m : \mathcal{X} \to \mathbb{R}$
  - zero, constant, polynomial...
- covariance function $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$
  - exponentiated quadratic $\kappa_{EQ}(\boldsymbol{x}, \boldsymbol{x}') = \sigma_f^2 \exp\left[-\frac{1}{2}\sum_i \frac{(x_i - x_i')^2}{\ell_i^2}\right]$

Gaussian Processes (GPs)

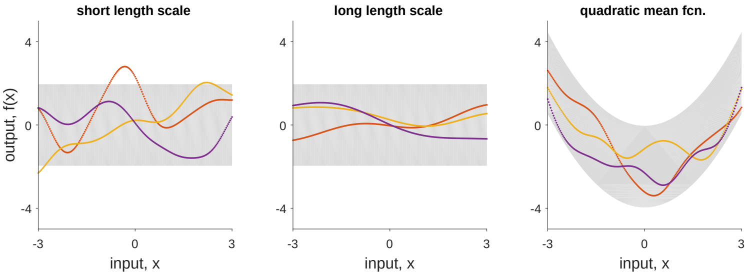
GPs used as *priors* over $f : \mathcal{X} \subseteq \mathbb{R}^D \to \mathbb{R}$

- mean function $m : \mathcal{X} \to \mathbb{R}$
  - zero, constant, polynomial...
- covariance function $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$
  - exponentiated quadratic $\kappa_{EQ}(\boldsymbol{x}, \boldsymbol{x}') = \sigma_f^2 \exp\left[-\frac{1}{2}\sum_i \frac{(x_i - x_i')^2}{\ell_i^2}\right]$
- observation function
  - Gaussian ($\sim$ small numerical noise σ^2_{obs})

Gaussian Processes (GPs)

GPs used as *priors* over $f : \mathcal{X} \subseteq \mathbb{R}^D \to \mathbb{R}$

- mean function $m : \mathcal{X} \to \mathbb{R}$
  - zero, constant, polynomial...
- covariance function $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$
  - exponentiated quadratic $\kappa_{EQ}(\boldsymbol{x}, \boldsymbol{x}') = \sigma_f^2 \exp\left[-\frac{1}{2}\sum_i \frac{(x_i - x_i')^2}{\ell_i^2}\right]$
- observation function
  - Gaussian ($\sim$ small numerical noise σ^2_{obs})



Posterior GPs

Training inputs $\mathbf{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$
Observed values $\boldsymbol{y} = (y_1 = f(\boldsymbol{x}_1), \ldots, y_n = f(\boldsymbol{x}_n))$
GP hyperparameters $\boldsymbol{\psi} = (\sigma_f, \boldsymbol{\ell}, \sigma_{\text{obs}}, m_0, \ldots)$

Posterior GPs

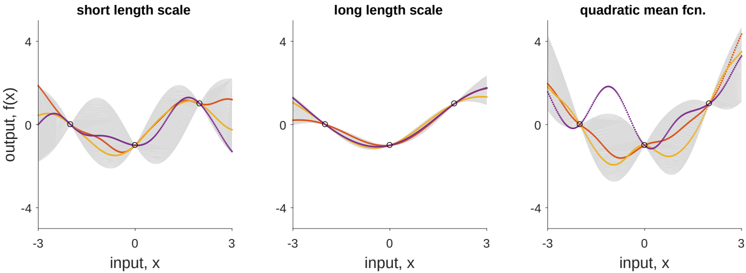
Training inputs $\mathbf{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$
Observed values $\boldsymbol{y} = (y_1 = f(\boldsymbol{x}_1), \ldots, y_n = f(\boldsymbol{x}_n))$
GP hyperparameters $\boldsymbol{\psi} = (\sigma_f, \boldsymbol{\ell}, \sigma_{\text{obs}}, m_0, \ldots)$

Posterior mean $\overline{f}(\mathbf{X}^*; \mathbf{X}, \boldsymbol{y}, \psi) = \kappa(\mathbf{X}, \mathbf{X}^*) \left[\kappa(\mathbf{X}, \mathbf{X}) + \sigma_{\text{obs}}^2 \mathbf{I}_n\right]^{-1} \boldsymbol{y}$
Posterior covariance $C(\mathbf{X}^*; \mathbf{X}, \boldsymbol{y}, \psi) =$ analytical expression

Posterior GPs

Training inputs $\mathbf{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$
Observed values $\boldsymbol{y} = (y_1 = f(\boldsymbol{x}_1), \ldots, y_n = f(\boldsymbol{x}_n))$
GP hyperparameters $\psi = (\sigma_f, \ell, \sigma_{\text{obs}}, m_0, \ldots)$

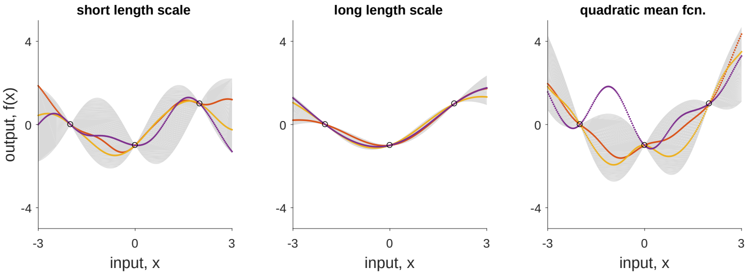
Posterior mean $\overline{f}(\mathbf{X}^*; \mathbf{X}, \boldsymbol{y}, \psi) = \kappa(\mathbf{X}, \mathbf{X}^*) \left[\kappa(\mathbf{X}, \mathbf{X}) + \sigma_{\text{obs}}^2 \mathbf{I}_n\right]^{-1} \boldsymbol{y}$
Posterior covariance $C(\mathbf{X}^*; \mathbf{X}, \boldsymbol{y}, \psi) =$ analytical expression



Posterior GPs

Training inputs $\mathbf{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$
Observed values $\boldsymbol{y} = (y_1 = f(\boldsymbol{x}_1), \ldots, y_n = f(\boldsymbol{x}_n))$
GP hyperparameters $\psi = (\sigma_f, \ell, \sigma_{\text{obs}}, m_0, \ldots)$

Posterior mean $\overline{f}(\mathbf{X}^*; \mathbf{X}, \boldsymbol{y}, \psi) = \kappa(\mathbf{X}, \mathbf{X}^*) \left[\kappa(\mathbf{X}, \mathbf{X}) + \sigma^2_{\text{obs}} \mathbf{I}_n\right]^{-1} \boldsymbol{y}$
Posterior covariance $C(\mathbf{X}^*; \mathbf{X}, \boldsymbol{y}, \psi) =$ analytical expression



GP marginal likelihood $p(\boldsymbol{y}|\mathbf{X}, \psi)$

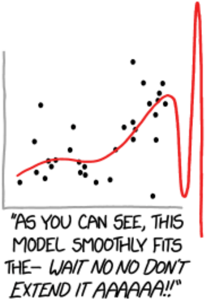
Why don't we use GPs *all the time*

Why don't we use GPs *all the time*

- Computation of $\left[\kappa(\mathbf{X},\mathbf{X}) + \sigma^2_{\text{obs}}\mathbf{I}_n\right]^{-1}$ is $O(n^3)$

Why don't we use GPs *all the time*

- Computation of $\left[\kappa(\mathbf{X}, \mathbf{X}) + \sigma^2_{\text{obs}} \mathbf{I}_n\right]^{-1}$ is $O(n^3)$
- Model mismatch



from `xkcd.com/2048`

Bayesian Quadrature (BQ)

Evaluate integral of (expensive) black-box functions

Bayesian Quadrature (BQ)

Evaluate integral of (expensive) black-box functions

$$Z = \int p(\boldsymbol{x}) f(\boldsymbol{x}) d\boldsymbol{x}$$

Bayesian Quadrature (BQ)

Evaluate integral of (expensive) black-box functions

$$Z = \int p(\boldsymbol{x}) f(\boldsymbol{x}) d\boldsymbol{x}$$

- $p(\boldsymbol{x})$ is Gaussian
- $f(\boldsymbol{x})$ approximated via a GP with EQ covariance

Bayesian Quadrature (BQ)

Evaluate integral of (expensive) black-box functions

$$Z = \int p(\boldsymbol{x}) f(\boldsymbol{x}) d\boldsymbol{x}$$

- $p(\boldsymbol{x})$ is Gaussian
- $f(\boldsymbol{x})$ approximated via a GP with EQ covariance

$\implies$ posterior Z can be computed analytically

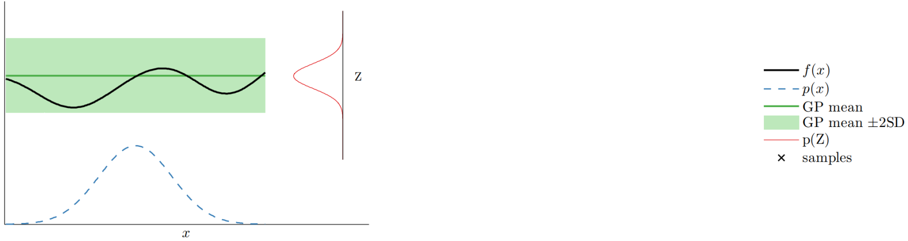
Bayesian Quadrature (BQ)

Evaluate integral of (expensive) black-box functions

$$Z = \int p(\boldsymbol{x}) f(\boldsymbol{x}) d\boldsymbol{x}$$

- $p(\boldsymbol{x})$ is Gaussian
- $f(\boldsymbol{x})$ approximated via a GP with EQ covariance

$\Longrightarrow$ posterior Z can be computed analytically



from Duvenaud, *NIPS workshop on Probabilistic Numerics* (2012)

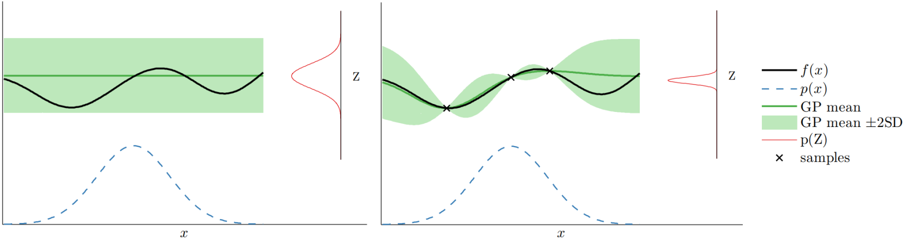
Bayesian Quadrature (BQ)

Evaluate integral of (expensive) black-box functions

$$Z = \int p(\mathbf{x}) f(\mathbf{x}) d\mathbf{x}$$

- $p(\mathbf{x})$ is Gaussian
- $f(\mathbf{x})$ approximated via a GP with EQ covariance

$\Longrightarrow$ posterior Z can be computed analytically



from Duvenaud, *NIPS workshop on Probabilistic Numerics* (2012)

BQ for Bayesian inference (previous work)

Evaluate marginal likelihood of (expensive) black-box functions

BQ for Bayesian inference (previous work)

Evaluate marginal likelihood of (expensive) black-box functions

- Bayesian Monte Carlo (BMC), Rasmussen and Ghahramani, *NIPS* (2003)
- Doubly-Bayesian quadrature (BBQ), Osborne et al., *NIPS* (2012)
- Warped seq. active Bayesian integration (WSABI), Gunter et al., *NIPS* (2014)

BQ for Bayesian inference (previous work)

Evaluate marginal likelihood of (expensive) black-box functions

- Bayesian Monte Carlo (BMC), Rasmussen and Ghahramani, *NIPS* (2003)
- Doubly-Bayesian quadrature (BBQ), Osborne et al., *NIPS* (2012)
- Warped seq. active Bayesian integration (WSABI), Gunter et al., *NIPS* (2014)

Active sampling

BQ for Bayesian inference (previous work)

Evaluate marginal likelihood of (expensive) black-box functions

- Bayesian Monte Carlo (BMC), Rasmussen and Ghahramani, *NIPS* (2003)
- Doubly-Bayesian quadrature (BBQ), Osborne et al., *NIPS* (2012)
- Warped seq. active Bayesian integration (WSABI), Gunter et al., *NIPS* (2014)

Active sampling

- Minimize expected variance of integral Z

BQ for Bayesian inference (previous work)

Evaluate marginal likelihood of (expensive) black-box functions

- Bayesian Monte Carlo (BMC), Rasmussen and Ghahramani, *NIPS* (2003)
- Doubly-Bayesian quadrature (BBQ), Osborne et al., *NIPS* (2012)
- Warped seq. active Bayesian integration (WSABI), Gunter et al., *NIPS* (2014)

Active sampling

- Minimize expected variance of integral Z
- *Uncertainty sampling*: Maximize variance of integrand $p(\boldsymbol{x})f(\boldsymbol{x})$

Putting things together

Putting things together

- Variational inference:

$$\begin{aligned} q_\phi(\boldsymbol{x}) &= \text{argmax}_\phi \text{ELBO}(\phi) \\ &= \text{argmax}_\phi \left\{ \int q_\phi(\boldsymbol{x}) \log \left[p(\mathcal{D}|\boldsymbol{x}) p(\boldsymbol{x}) \right] d\boldsymbol{x} + \mathcal{H}[q_\phi(\boldsymbol{x})] \right\} \end{aligned}$$

Putting things together

- Variational inference:

$$\begin{aligned} q_\phi(\boldsymbol{x}) &= \text{argmax}_\phi \text{ELBO}(\phi) \\ &= \text{argmax}_\phi \left\{ \int q_\phi(\boldsymbol{x}) \log\left[p(\mathcal{D}|\boldsymbol{x})p(\boldsymbol{x})\right] d\boldsymbol{x} + \mathcal{H}[q_\phi(\boldsymbol{x})] \right\} \end{aligned}$$

- Bayesian quadrature:

$$Z = \int q(\boldsymbol{x}) f(\boldsymbol{x}) d\boldsymbol{x}$$

Putting things together

- Variational inference:

$$
\begin{aligned}
q_\phi(\boldsymbol{x}) &= \text{argmax}_\phi \text{ELBO}(\phi) \\
&= \text{argmax}_\phi \left\{ \int q_\phi(\boldsymbol{x}) \log \left[p(\mathcal{D}|\boldsymbol{x}) p(\boldsymbol{x}) \right] d\boldsymbol{x} + \mathcal{H}[q_\phi(\boldsymbol{x})] \right\}
\end{aligned}
$$

- Bayesian quadrature:

$$
Z = \int q(\boldsymbol{x}) f(\boldsymbol{x}) d\boldsymbol{x}
$$

Putting things together

- Variational inference:

$$
\begin{aligned}
q_\phi(\boldsymbol{x}) &= \text{argmax}_\phi \text{ELBO}(\phi) \\
&= \text{argmax}_\phi \left\{ \int q_\phi(\boldsymbol{x}) \log \left[p(\mathcal{D}|\boldsymbol{x}) p(\boldsymbol{x}) \right] d\boldsymbol{x} + \mathcal{H}[q_\phi(\boldsymbol{x})] \right\}
\end{aligned}
$$

- Bayesian quadrature:

$$
Z = \int q(\boldsymbol{x}) f(\boldsymbol{x}) d\boldsymbol{x}
$$

VI + BQ = VBMC

1 Introduction and motivation

2 Background Tools

3 Variational Bayesian Monte Carlo

4 Experiments

VBMC in an nutshell

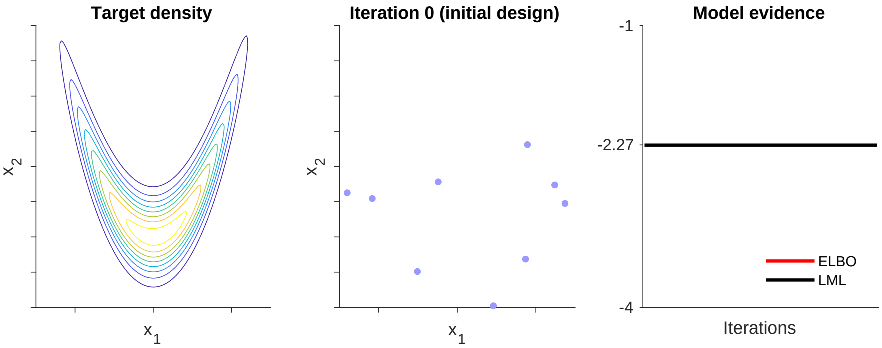
In each iteration t:

1. (Actively) sample new points, evaluate $f = \log p(\mathcal{D}|\boldsymbol{x}_{\text{new}})p(\boldsymbol{x}_{\text{new}})$
2. train GP model of the log joint f
3. update variational posterior q_{ϕ_t} by optimizing the ELBO

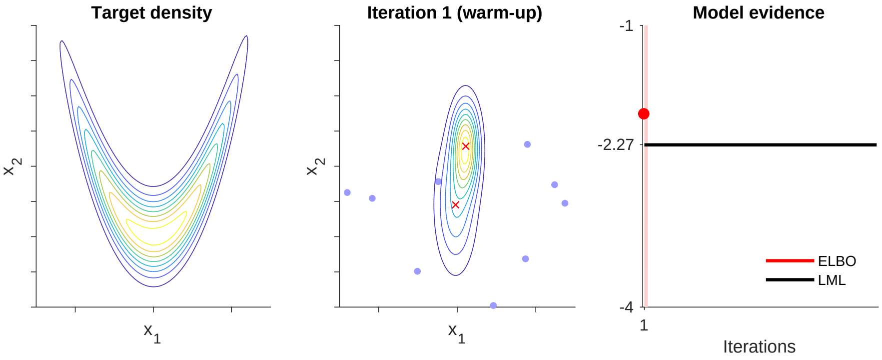
Loop until reaching termination criterion

Acerbi, *NeurIPS* (2018)

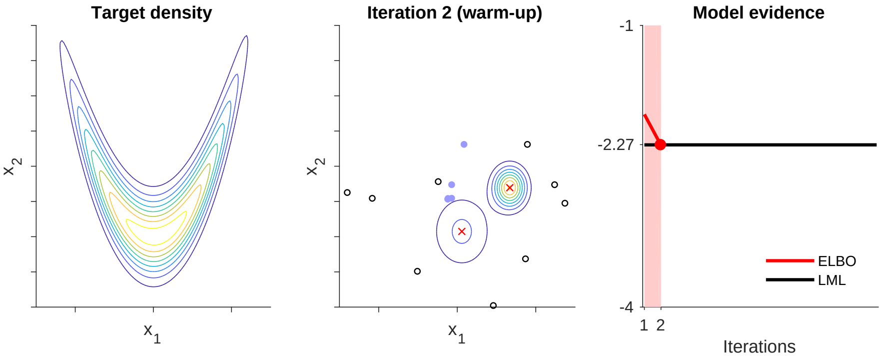
VBMC demo


Target density
x 2
x 1
Iteration 0 (initial design)
x 2
x 1
Model evidence
-1
-2.27
-4
ELBO
LML
Iterations

VBMC demo

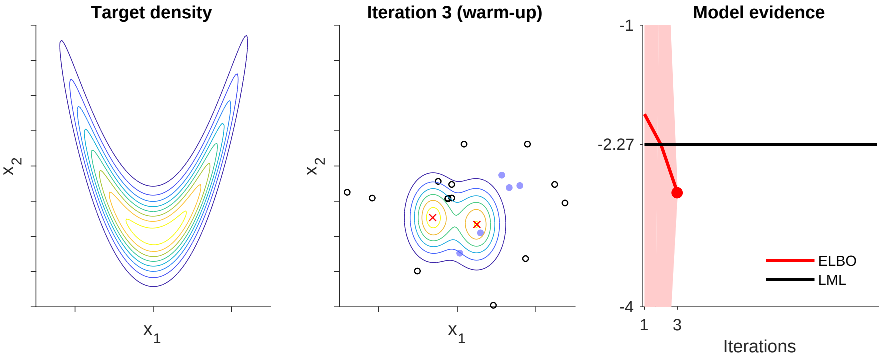

Target density
Iteration 1 (warm-up)
Model evidence
x_2
x_1
x_2
x_1
-1
-2.27
-4
1
Iterations
ELBO
LML

VBMC demo



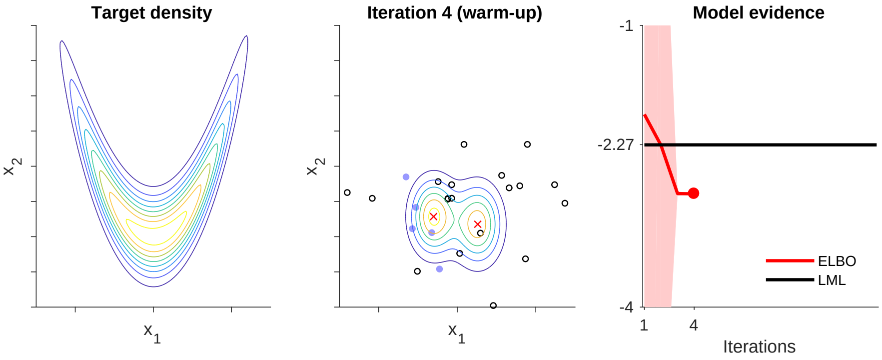
Target density
x2
x1
Iteration 2 (warm-up)
x2
x1
Model evidence
-1
-2.27
-4
1 2
Iterations
ELBO
LML

VBMC demo



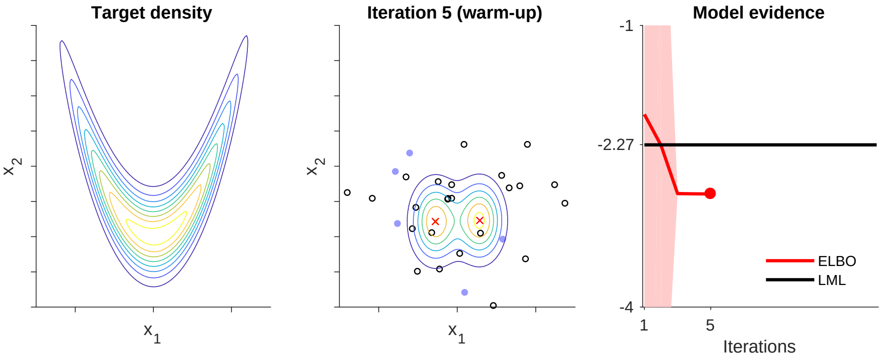
Target density
Iteration 3 (warm-up)
Model evidence
x2
x1
x2
x1
-1
-2.27
-4
1
3
Iterations
ELBO
LML

VBMC demo



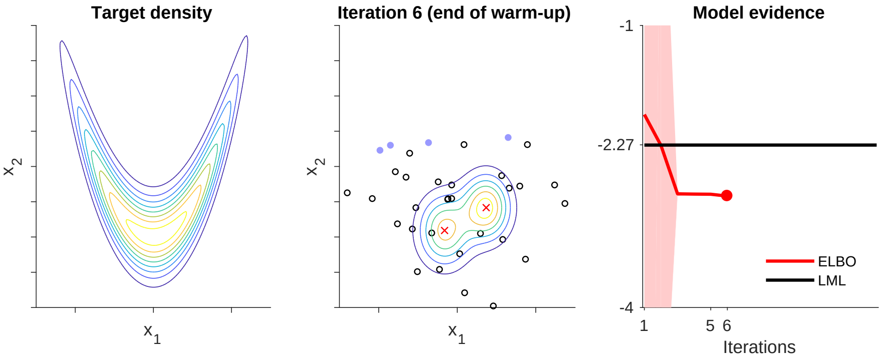
Target density
x 2
x 1
Iteration 4 (warm-up)
x 2
x 1
Model evidence
-1
-2.27
-4
1
4
Iterations
ELBO
LML

VBMC demo



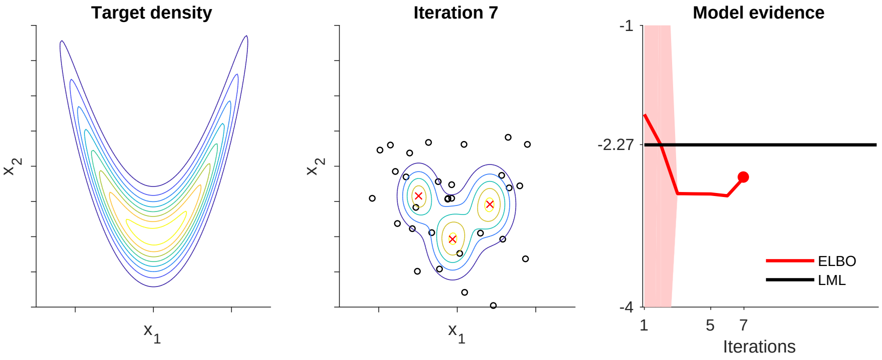
Target density
Iteration 5 (warm-up)
Model evidence
x_2
x_1
-1
-2.27
-4
1
5
Iterations
ELBO
LML

VBMC demo



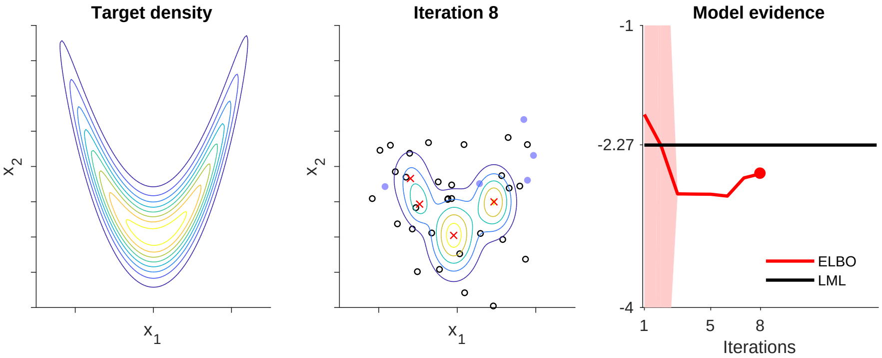
Target density
x2
x1
Iteration 6 (end of warm-up)
x2
x1
Model evidence
-1
-2.27
-4
1
5 6
Iterations
ELBO
LML

VBMC demo



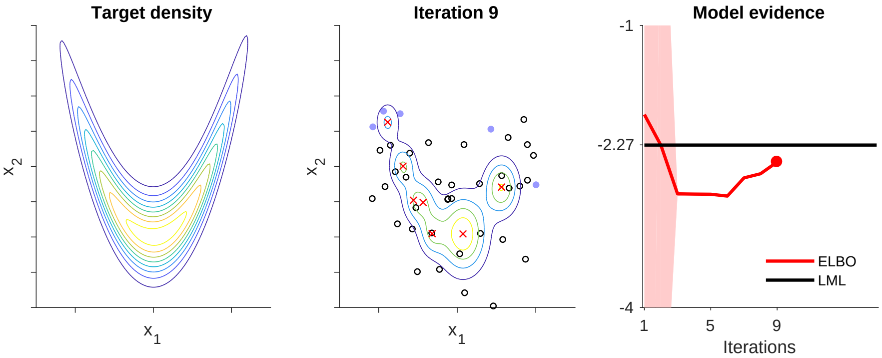
Target density
x 2
x 1
Iteration 7
x 2
x 1
Model evidence
-1
-2.27
-4
1
5
7
Iterations
ELBO
LML

VBMC demo



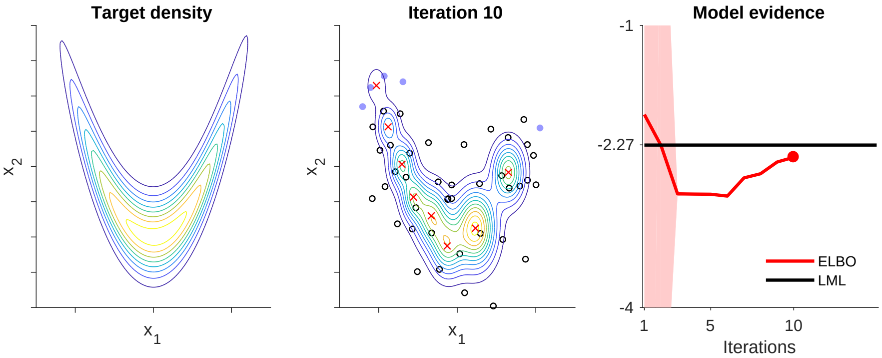
Target density
x_2
x_1
Iteration 8
x_2
x_1
Model evidence
-1
-2.27
-4
1
5
8
Iterations
ELBO
LML

VBMC demo



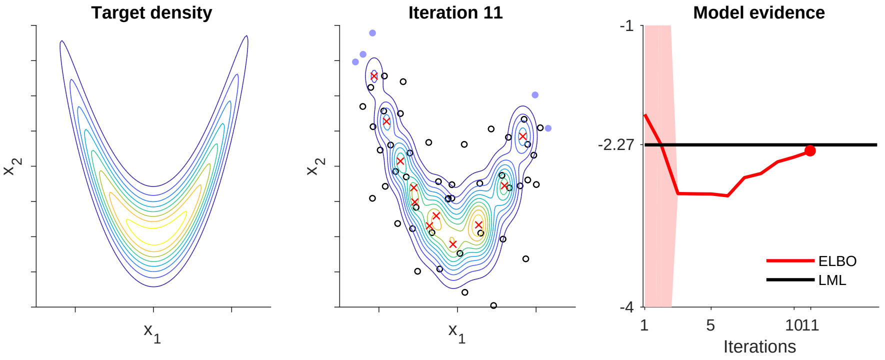
Target density
x_2
x_1
Iteration 9
x_2
x_1
Model evidence
-1
-2.27
-4
1
5
9
Iterations
ELBO
LML

VBMC demo



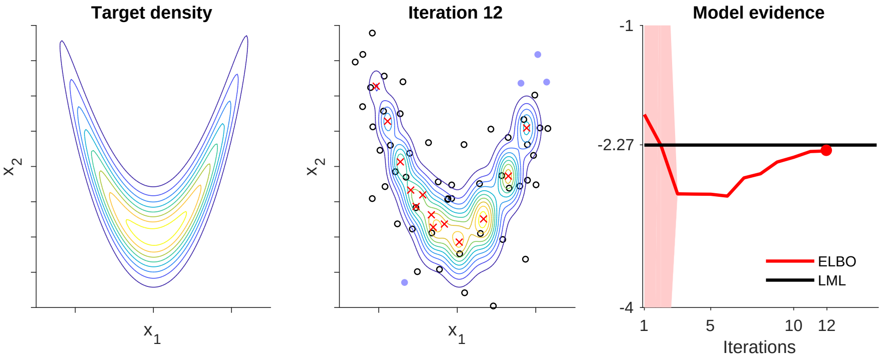
Target density
x2
x1
Iteration 10
x2
x1
Model evidence
-1
-2.27
-4
1
5
10
Iterations
ELBO
LML

VBMC demo



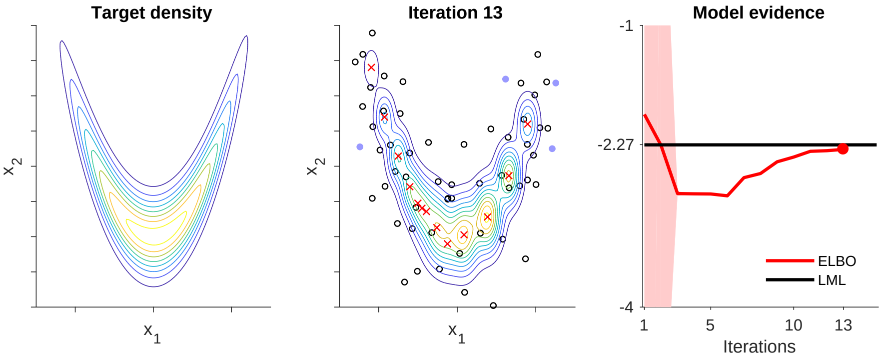
Target density
x_2
x_1
Iteration 11
x_2
x_1
Model evidence
-1
-2.27
-4
1
5
1011
Iterations
ELBO
LML

VBMC demo



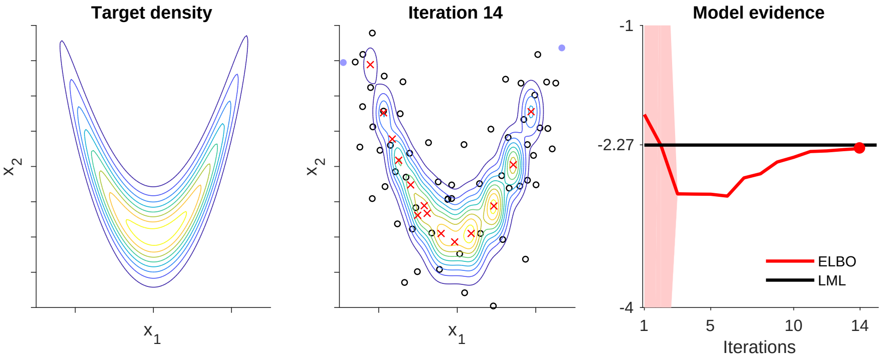
Target density
x_2
x_1
Iteration 12
x_2
x_1
Model evidence
-1
-2.27
-4
1
5
10
12
Iterations
ELBO
LML

VBMC demo



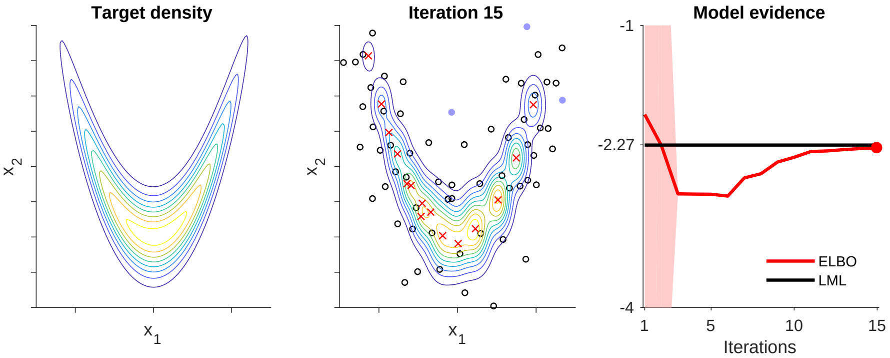
Target density
x 2
x 1
Iteration 13
x 2
x 1
Model evidence
-1
-2.27
-4
1
5
10
13
Iterations
ELBO
LML

VBMC demo



Target density
x1
x2
Iteration 14
x1
x2
Model evidence
-1
-2.27
-4
1
5
10
14
Iterations
ELBO
LML

VBMC demo


Target density
x_2
x_1
Iteration 15
x_2
x_1
Model evidence
-1
-2.27
-4
1
5
10
15
Iterations
ELBO
LML

Variational posterior

$$q_\phi(\boldsymbol{x}) = \sum_{k=1}^{K} w_k \mathcal{N}\left(\boldsymbol{x}; \boldsymbol{\mu}_k, \sigma_k^2 \boldsymbol{\Sigma}\right), \quad \boldsymbol{\Sigma} \equiv \mathrm{diag}[{\lambda^{(1)}}^2, \ldots, {\lambda^{(D)}}^2]$$

Variational posterior

$$q_\phi(\boldsymbol{x}) = \sum_{k=1}^{K} w_k \mathcal{N}\left(\boldsymbol{x}; \boldsymbol{\mu}_k, \sigma_k^2 \boldsymbol{\Sigma}\right), \quad \boldsymbol{\Sigma} \equiv \mathrm{diag}[{\lambda^{(1)}}^2, \ldots, {\lambda^{(D)}}^2]$$

- $\boldsymbol{x} \in \mathbb{R}^D$
- $\phi \equiv (w_1, \ldots, w_K, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K, \sigma_1, \ldots, \sigma_K, \boldsymbol{\lambda})$
- $K(D+2)+D$ parameters
- K is changed adaptively each iteration

Gaussian process representation

$$f(\boldsymbol{x}) = \log p(\mathcal{D}|\boldsymbol{x})p(\boldsymbol{x})$$

Gaussian process representation

$$f(\boldsymbol{x}) = \log p(\mathcal{D}|\boldsymbol{x})p(\boldsymbol{x})$$

- Exponentiated quadratic covariance
- Gaussian observation noise
- *Negative quadratic* mean

Gaussian process representation

$$f(\boldsymbol{x}) = \log p(\mathcal{D}|\boldsymbol{x})p(\boldsymbol{x})$$

- Exponentiated quadratic covariance
- Gaussian observation noise
- *Negative quadratic* mean

$$m_{\mathrm{NQ}}(\boldsymbol{x}) = m_0 - \frac{1}{2}\sum_{i=1}^{D} \frac{\left(x^{(i)} - x_{\mathrm{m}}^{(i)}\right)^2}{\omega^{(i)^2}},$$

Gaussian process representation

$$f(\boldsymbol{x}) = \log p(\mathcal{D}|\boldsymbol{x})p(\boldsymbol{x})$$

- Exponentiated quadratic covariance
- Gaussian observation noise
- *Negative quadratic* mean

$$m_{\mathrm{NQ}}(\boldsymbol{x}) = m_0 - \frac{1}{2}\sum_{i=1}^{D} \frac{\left(x^{(i)} - x_{\mathrm{m}}^{(i)}\right)^2}{\omega^{(i)2}},$$

Sample over GP hyperparameters (later optimize)

Variational optimization

$$\mathrm{ELBO}(\phi, f) = \int q_{\phi}(\boldsymbol{x}) f(\boldsymbol{x}) d\boldsymbol{x} + \mathcal{H}[q_{\phi}(\boldsymbol{x})]$$

Variational optimization

$$\text{ELBO}(\phi, f) = \int q_\phi(\boldsymbol{x}) f(\boldsymbol{x}) d\boldsymbol{x} + \mathcal{H}[q_\phi(\boldsymbol{x})]$$

- Expected log joint and gradient are analytical

Variational optimization

$$\text{ELBO}(\phi, f) = \int q_\phi(\boldsymbol{x}) f(\boldsymbol{x}) d\boldsymbol{x} + \mathcal{H}[q_\phi(\boldsymbol{x})]$$

- Expected log joint and gradient are analytical
- Entropy via simple Monte Carlo
- Entropy gradient via reparametrization trick (Kingma & Welling, 2013; Miller et al., 2017)

Variational optimization

$$\text{ELBO}(\phi, f) = \int q_\phi(\boldsymbol{x}) f(\boldsymbol{x}) d\boldsymbol{x} + \mathcal{H}[q_\phi(\boldsymbol{x})]$$

- Expected log joint and gradient are analytical
- Entropy via simple Monte Carlo
- Entropy gradient via reparametrization trick (Kingma & Welling, 2013; Miller et al., 2017)

Optimize with SGD (Adam; Kingma & Ba, 2014)

Active sampling

Optimize *acquisition function*: $\boldsymbol{x}_{\text{next}} = \arg\max_{\boldsymbol{x}} a(\boldsymbol{x})$

Active sampling

Optimize *acquisition function*: $\boldsymbol{x}_{\text{next}} = \arg\max_{\boldsymbol{x}} a(\boldsymbol{x})$

Goal: Evaluate $\mathbb{E}_{\phi}\left[f\right] = \int q_{\phi}(\boldsymbol{x}) f(\boldsymbol{x}) d\boldsymbol{x}$

Active sampling

Optimize *acquisition function*: $\boldsymbol{x}_{\text{next}} = \arg\max_{\boldsymbol{x}} a(\boldsymbol{x})$

Goal: Evaluate $\mathbb{E}_{\phi}[f] = \int q_{\phi}(\boldsymbol{x}) f(\boldsymbol{x}) d\boldsymbol{x}$

$\Longrightarrow$ 'Vanilla' uncertainty sampling: $a_{\text{us}}(\boldsymbol{x}) = V(\boldsymbol{x}) q_{\phi}(\boldsymbol{x})^2$

Active sampling

Optimize *acquisition function*: $\boldsymbol{x}_{\text{next}} = \arg\max_{\boldsymbol{x}} a(\boldsymbol{x})$

Goal: Evaluate $\mathbb{E}_{\phi}[f] = \int q_{\phi}(\boldsymbol{x}) f(\boldsymbol{x}) d\boldsymbol{x}$

$\Longrightarrow$ 'Vanilla' uncertainty sampling: $a_{\text{us}}(\boldsymbol{x}) = V(\boldsymbol{x}) q_{\phi}(\boldsymbol{x})^2$

Goal: Evaluate $\mathbb{E}_{\phi_1}[f], \mathbb{E}_{\phi_2}[f], \ldots, \mathbb{E}_{\phi_T}[f]$

Active sampling

Optimize *acquisition function*: $\boldsymbol{x}_{\text{next}} = \arg\max_{\boldsymbol{x}} a(\boldsymbol{x})$

Goal: Evaluate $\mathbb{E}_{\phi}\left[f\right] = \int q_{\phi}(\boldsymbol{x}) f(\boldsymbol{x}) d\boldsymbol{x}$

$\Longrightarrow$ 'Vanilla' uncertainty sampling: $a_{\text{us}}(\boldsymbol{x}) = V(\boldsymbol{x}) q_{\phi}(\boldsymbol{x})^2$

Goal: Evaluate $\mathbb{E}_{\phi_1}\left[f\right], \mathbb{E}_{\phi_2}\left[f\right], \ldots, \mathbb{E}_{\phi_T}\left[f\right]$

$\Longrightarrow$ Prospective uncertainty sampling: $a_{\text{pro}}(\boldsymbol{x}) = V(\boldsymbol{x}) q_{\phi}(\boldsymbol{x}) \exp\left(\overline{f}(\boldsymbol{x})\right)$

Algorithmic details

Algorithmic details

Evidence Lower Confidence Bound (ELCBO)

$$\mathrm{ELCBO}(\phi, f) = \mathrm{ELBO}(\phi, f) - \beta_{LCB} \cdot \mathrm{SD}\left[\mathbb{E}_{\phi}\left[f\right]\right]$$

Algorithmic details

Evidence Lower Confidence Bound (ELCBO)

$$\mathrm{ELCBO}(\phi, f) = \mathrm{ELBO}(\phi, f) - \beta_{LCB} \cdot \mathrm{SD}\left[\mathbb{E}_{\phi}\left[f\right]\right]$$

- Adaptive number of components
  - Try adding new components at each iteration
  - Prune small components with little effect on ELCBO

Algorithmic details

Evidence Lower Confidence Bound (ELCBO)

$$\mathrm{ELCBO}(\phi, f) = \mathrm{ELBO}(\phi, f) - \beta_{LCB} \cdot \mathrm{SD}\left[\mathbb{E}_{\phi}\left[f\right]\right]$$

- Adaptive number of components
  - Try adding new components at each iteration
  - Prune small components with little effect on ELCBO
- Warm-up
  - Clamp $K = 2$, $w_1 = w_2 = \frac{1}{2}$
  - Warm-up ends when ELCBO improvement slows down

Algorithmic details

Evidence Lower Confidence Bound (ELCBO)

$$\mathrm{ELCBO}(\phi, f) = \mathrm{ELBO}(\phi, f) - \beta_{LCB} \cdot \mathrm{SD}\left[\mathbb{E}_{\phi}\left[f\right]\right]$$

- Adaptive number of components
  - Try adding new components at each iteration
  - Prune small components with little effect on ELCBO
- Warm-up
  - Clamp $K = 2$, $w_1 = w_2 = \frac{1}{2}$
  - Warm-up ends when ELCBO improvement slows down
- Termination criteria
  - Reliability index $\rho(t)$
  - Long-term stability: $\rho(t) \leq 1$ for n_{stable} iterations

1 Introduction and motivation

2 Background Tools

3 Variational Bayesian Monte Carlo

4 Experiments

Experiment setup

Benchmark sets:

- Three families of synthetic functions ($D \in \{2, 4, 6, 8, 10\}$)
- Neuronal model with real data ($D = 7$)

Experiment setup

Benchmark sets:

- Three families of synthetic functions ($D \in \{2, 4, 6, 8, 10\}$)
- Neuronal model with real data ($D = 7$)

Procedure:

- Budget of $50 \times (D + 2)$ likelihood evaluations
- Metrics
  - Error wrt true log marginal likelihood (LML)
  - ‘Gaussianized’ symmetrized KL divergence between ground truth and posterior approximation (gsKL)

Algorithms

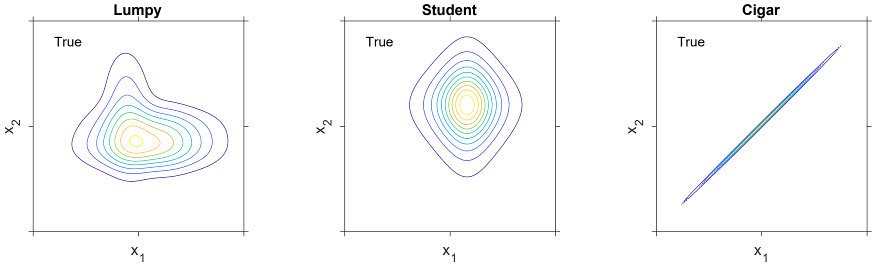
- VBMC-U (a_{us}) and VBMC-P (a_{pro})
- Simple Monte Carlo (SMC), annealed importance sampling (AIS)
- Bayesian Monte Carlo (BMC)
- Doubly-Bayesian quadrature (BBQ, BBQ*)
- WSABI, linearized (WSABI-L) and moment-matching (WSABI-M)
- Posterior estimation via GPs (AGP, BAPE)

Algorithms

- VBMC-U (a_{us}) and VBMC-P (a_{pro})
- Simple Monte Carlo (SMC), annealed importance sampling (AIS)
- Bayesian Monte Carlo (BMC)
- Doubly-Bayesian quadrature (BBQ, BBQ*)
- WSABI, linearized (WSABI-L) and moment-matching (WSABI-M)
- Posterior estimation via GPs (AGP, BAPE)

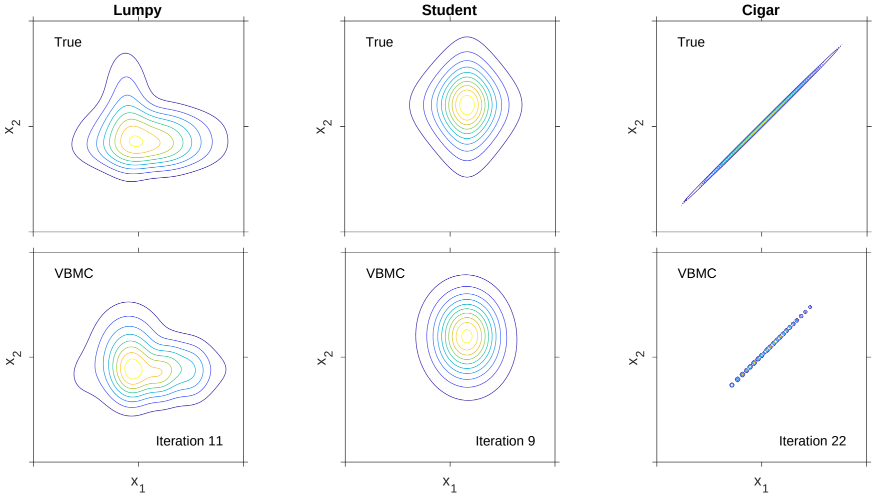
Synthetic target densities

Three families: *Lumpy*, *Student*, *Cigar* $\qquad D \in \{2, 4, 6, 8, 10\}$

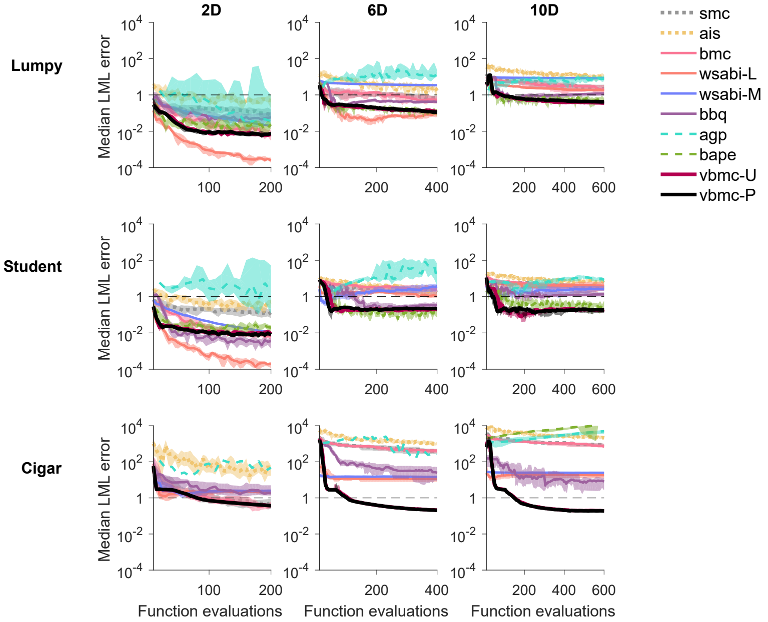


Synthetic target densities

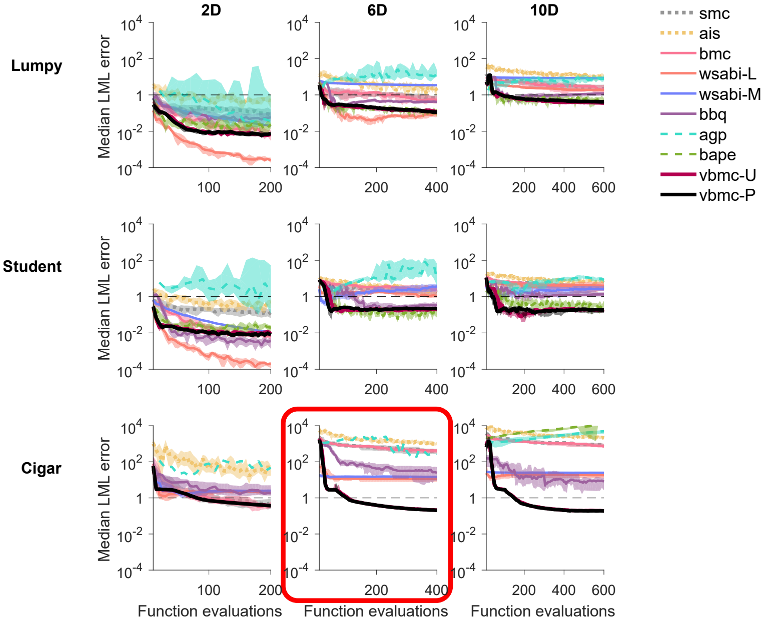
Three families: *Lumpy*, *Student*, *Cigar* $\qquad D \in \{2, 4, 6, 8, 10\}$



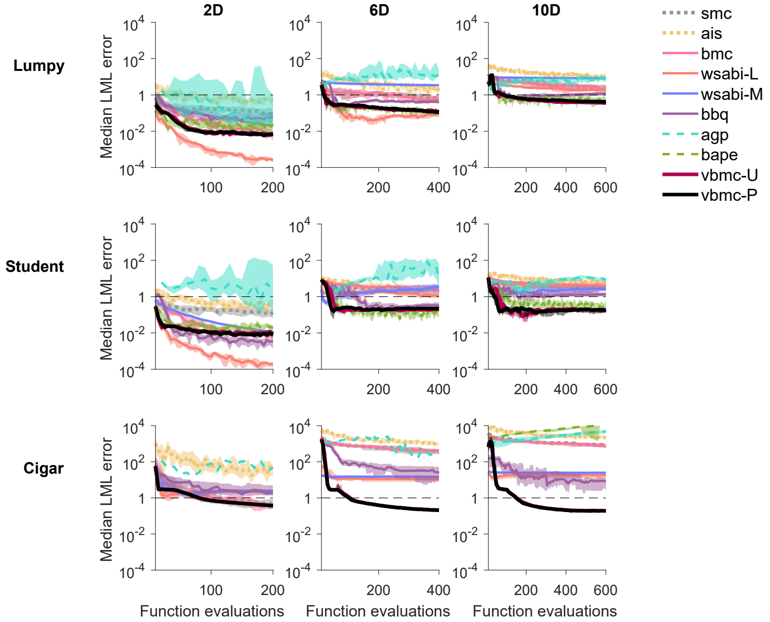
Synthetic target densities: Results



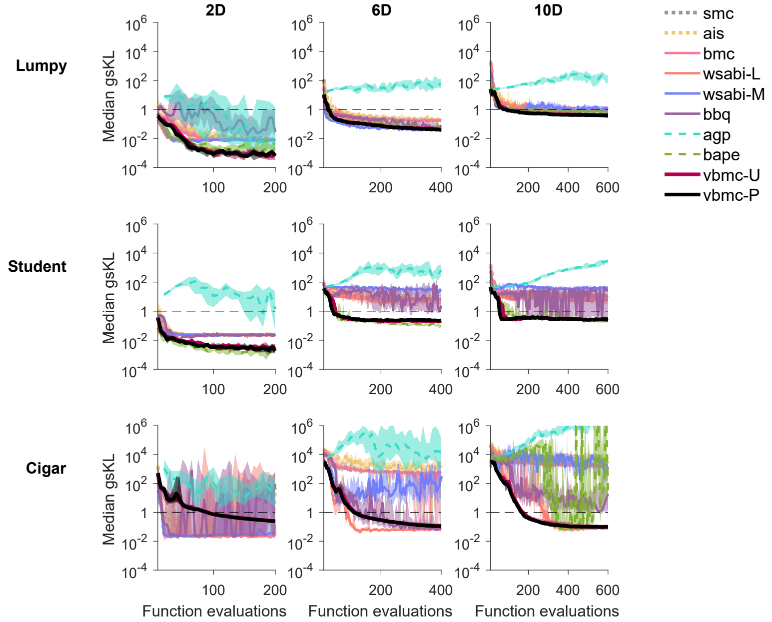
Synthetic target densities: Results



Synthetic target densities: Results

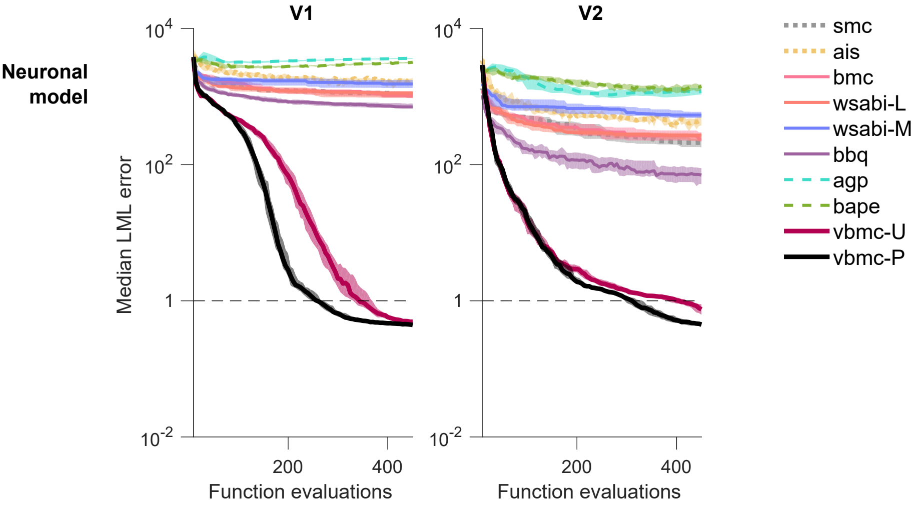


Synthetic target densities: Results



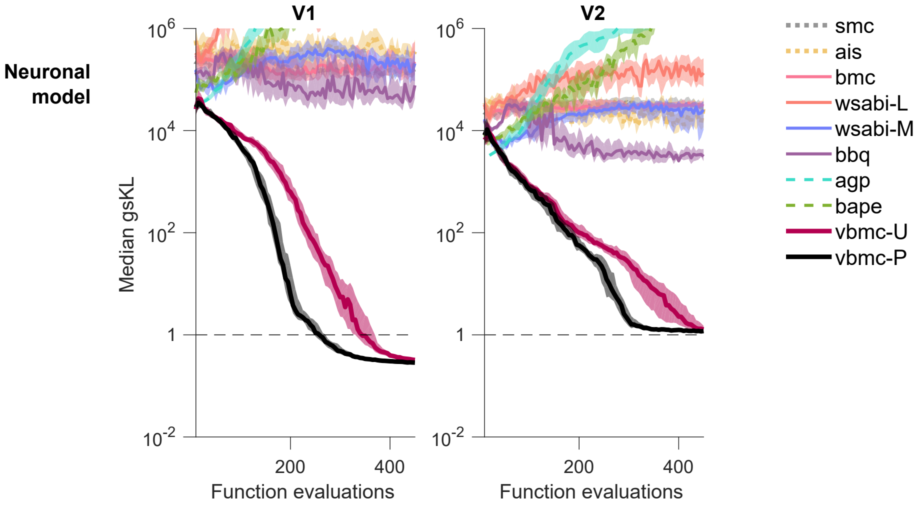
Neuronal model: Results

Two datasets: V1, V2 $\qquad D = 7$

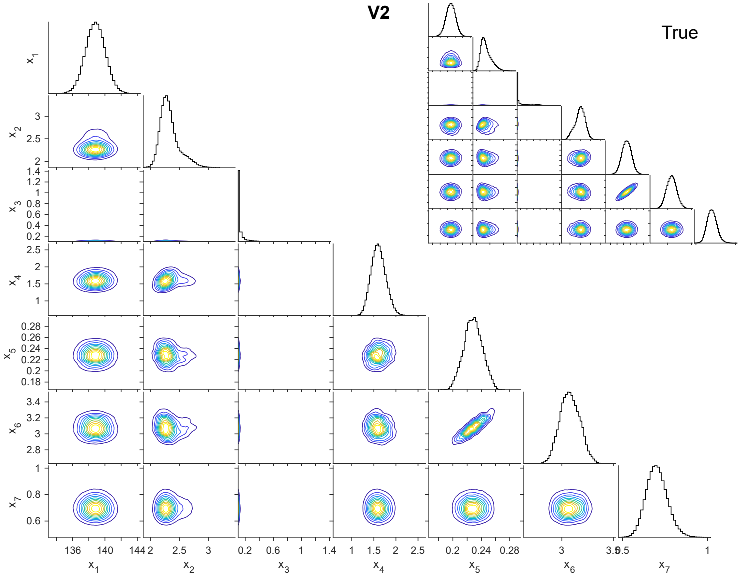


Neuronal model: Results

Two datasets: V1, V2 $\quad\quad D = 7$

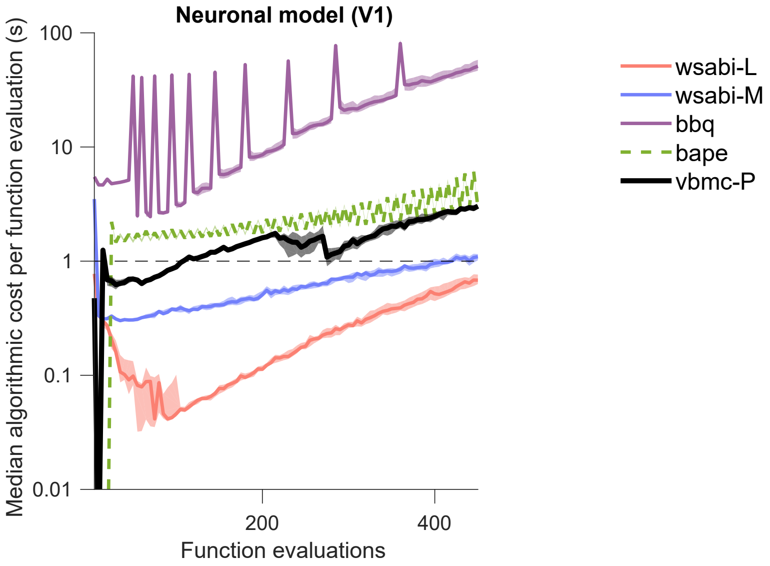


Neuronal model: VBMC



VBMC (iteration 52)

Computational cost



What's the secret sauce?

What's the secret sauce?

Other quadrature methods: (BMC, BBQ, WSABI)

$$Z = \int p(\boldsymbol{x}) p(\mathcal{D}|\boldsymbol{x}) d\boldsymbol{x}$$

What's the secret sauce?

Other quadrature methods: (BMC, BBQ, WSABI)

$$Z = \int p(\boldsymbol{x}) p(\mathcal{D}|\boldsymbol{x}) d\boldsymbol{x}$$

VBMC:

$$\mathcal{I}_k = \int q_k(\boldsymbol{x}) \log \left[p(\boldsymbol{x}) p(\mathcal{D}|\boldsymbol{x}) \right] d\boldsymbol{x}$$

What's the secret sauce?

Other quadrature methods: (BMC, BBQ, WSABI)

$$Z = \int p(\boldsymbol{x}) p(\mathcal{D}|\boldsymbol{x}) d\boldsymbol{x}$$

VBMC:

$$\mathcal{I}_k = \int q_k(\boldsymbol{x}) \log \left[p(\boldsymbol{x}) p(\mathcal{D}|\boldsymbol{x}) \right] d\boldsymbol{x}$$

- GP representation

What's the secret sauce?

Other quadrature methods: (BMC, BBQ, WSABI)

$$Z = \int p(\mathbf{x}) p(\mathcal{D}|\mathbf{x}) d\mathbf{x}$$

VBMC:

$$\mathcal{I}_k = \int q_k(\mathbf{x}) \log \left[p(\mathbf{x}) p(\mathcal{D}|\mathbf{x}) \right] d\mathbf{x}$$

- GP representation
- Integration scope

Discussion and future directions

- Model mismatch and robustness (e.g., nonstationarity)

Discussion and future directions

- Model mismatch and robustness (e.g., nonstationarity)
- Alternative GP representations

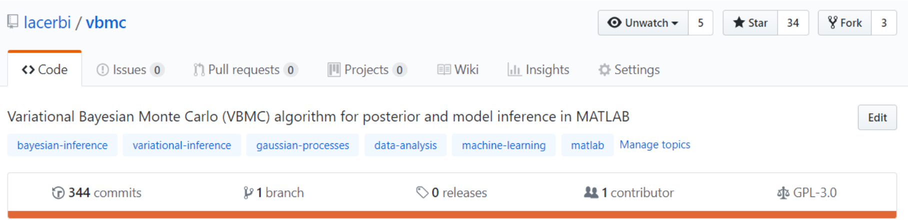
Discussion and future directions

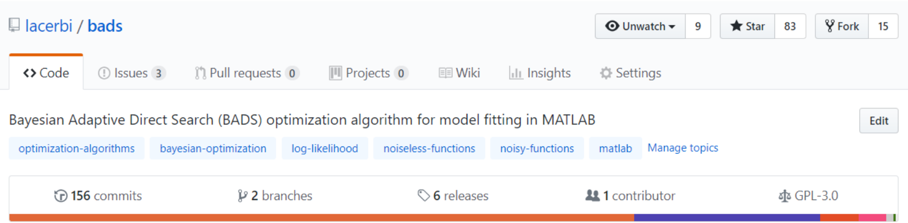
- Model mismatch and robustness (e.g., nonstationarity)
- Alternative GP representations
- More principled algorithmic solutions

Discussion and future directions

- Model mismatch and robustness (e.g., nonstationarity)
- Alternative GP representations
- More principled algorithmic solutions
- Killer application in machine learning

Toolboxes


lacerbi / vbmc
Unwatch 5
Star 34
Fork 3
Code
Issues 0
Pull requests 0
Projects 0
Wiki
Insights
Settings
Variational Bayesian Monte Carlo (VBMC) algorithm for posterior and model inference in MATLAB
Edit
bayesian-inference
variational-inference
gaussian-processes
data-analysis
machine-learning
matlab
Manage topics
344 commits
1 branch
0 releases
1 contributor
GPL-3.0


lacerbi / bads
Unwatch 9
Star 83
Fork 15
Code
Issues 3
Pull requests 0
Projects 0
Wiki
Insights
Settings
Bayesian Adaptive Direct Search (BADS) optimization algorithm for model fitting in MATLAB
Edit
optimization-algorithms
bayesian-optimization
log-likelihood
noiseless-functions
noisy-functions
matlab
Manage topics
156 commits
2 branches
6 releases
1 contributor
GPL-3.0

Final slide

- VBMC paper: `https://arxiv.org/abs/1810.05558`
- VBMC toolbox at: `github.com/lacerbi/vbmc`
- BADS toolbox at: `github.com/lacerbi/bads`

Final slide

- VBMC paper: `https://arxiv.org/abs/1810.05558`
- VBMC toolbox at: `github.com/lacerbi/vbmc`
- BADS toolbox at: `github.com/lacerbi/bads`

Acknowledgments

- Alexandre Pouget and the Pouget lab
- Robbe Goris

Final slide

- VBMC paper: `https://arxiv.org/abs/1810.05558`
- VBMC toolbox at: `github.com/lacerbi/vbmc`
- BADS toolbox at: `github.com/lacerbi/bads`

Acknowledgments

- Alexandre Pouget and the Pouget lab
- Robbe Goris

Thanks!

Control experiment

LML computed with WSABI-L on VBMC samples