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
Bayesian inference with expensive black-box statistical models
Goal

Bayesian inference with expensive black-box statistical models

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

Why Bayesian inference?

Luigi Acerbi
VBMC
Nov 26, 2018
Goal

Bayesian inference with expensive **black-box** statistical models

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

Bayesian inference with expensive black-box statistical models

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

**Bayesian inference with expensive black-box statistical models**

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

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

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

**Bayesian inference with expensive black-box statistical models**

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

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

Marginal likelihood: $p(D) = \int p(D|x)p(x)dx$

(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 $p(x|D)$ with $q_\phi(x)$
Variational inference

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

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

$\Rightarrow$ Maximize $\text{ELBO}(\phi) = \mathbb{E}_{q_\phi} \left[ \log p(D|x) p(x) \right] + \mathcal{H}[q_\phi(x)]$

- $\mathbb{E}_{q_\phi}$ expected log joint
- $\mathcal{H}[q_\phi(x)]$ entropy
Variational inference

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

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

\[\text{expected log joint} \quad \text{entropy}\]
Variational inference

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

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

Obtains
- An approximate posterior $q_\phi(x)$
- A lower bound to the log marginal likelihood, ELBO($\phi$)
Variational inference

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

\[ \Rightarrow \text{Maximize ELBO}(\phi) = \mathbb{E}_{q_\phi} \left[ \log p(\mathcal{D}|\mathbf{x})p(\mathbf{x}) \right] + \mathcal{H}[q_\phi(\mathbf{x})] \leq \log p(\mathcal{D}) \]

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

VI casts Bayesian inference into optimization + integration
Variational inference: example
Variational inference: example

\[ q_\phi(x) = \mathcal{N}(x, \mu, \sigma^2) \quad \quad \phi = (\mu, \sigma^2) \]
Variational inference: example

\[ q_\phi(x) = \sum_{k=1}^{K} w_k \mathcal{N}(x, \mu_k, \sigma_k^2) \]

\[ \phi = (w_k, \mu_k, \sigma_k^2)_{k=1}^K \]
Variational inference: example

\[ q_\phi(x) = \sum_{k=1}^{K} w_k \mathcal{N}(x, \mu_k, \sigma_k^2) \]

\[ \phi = (w_k, \mu_k, \sigma_k^2)_{k=1}^{K} \]
Gaussian Processes (GPs)
GPs used as *priors* over $f : \mathcal{X} \subseteq \mathbb{R}^D \rightarrow \mathbb{R}$
Gaussian Processes (GPs)

GPs used as *priors* over \( f : \mathcal{X} \subseteq \mathbb{R}^D \rightarrow \mathbb{R} \)

- mean function \( m : \mathcal{X} \rightarrow \mathbb{R} \)
  - 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 $\kappa_{EQ}(x, x') = \sigma_f^2 \exp \left[ -\frac{1}{2} \sum_i \frac{(x_i - x'_i)^2}{\ell_i^2} \right]$
Gaussian Processes (GPs)

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

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

---

![Graphs showing short and long length scales with quadratic mean function.](image-url)
Posterior GPs

Training inputs $\mathbf{X} = (x_1, \ldots, x_n)$
Observed values $\mathbf{y} = (y_1 = f(x_1), \ldots, y_n = f(x_n))$
GP hyperparameters $\psi = (\sigma_f, \ell, \sigma_{\text{obs}}, m_0, \ldots)$
Posterior GPs

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

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

Training inputs $\mathbf{X} = (x_1, \ldots, x_n)$
Observed values $\mathbf{y} = (y_1 = f(x_1), \ldots, y_n = f(x_n))$
GP hyperparameters $\psi = (\sigma_f, \ell, \sigma_{\text{obs}}, m_0, \ldots)$

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

Training inputs $\mathbf{X} = (x_1, \ldots, x_n)$
Observed values $\mathbf{y} = (y_1 = f(x_1), \ldots, y_n = f(x_n))$
GP hyperparameters $\mathbf{\psi} = (\sigma_f, \ell, \sigma_{obs}, m_0, \ldots)$

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

GP marginal likelihood $p(\mathbf{y} | \mathbf{X}, \mathbf{\psi})$
Why don’t we use GPs *all the time*
Why don’t we use GPs all the time

- Computation of \[ [\kappa(X, X) + \sigma_{\text{obs}}^2 I_n]^{-1} \] is \( O(n^3) \)
Why don’t we use GPs *all the time*

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

"As you can see, this model smoothly fits the—wait no no don't extend it aaaaaa!!"

from xkcd.com/2048
Bayesian Quadrature (BQ)

Evaluate integral of (expensive) black-box functions
Bayesian Quadrature (BQ)

Evaluate integral of (expensive) black-box functions

\[ Z = \int p(x)f(x)dx \]

\( p(x) \) is Gaussian
\( f(x) \) approximated via a GP with EQ covariance
\Rightarrow posterior
\( Z \) can be computed analytically

Duvenaud, NIPS workshop on Probabilistic Numerics (2012)
Bayesian Quadrature (BQ)

Evaluate integral of (expensive) black-box functions

\[ Z = \int p(x)f(x)dx \]

- \( p(x) \) is Gaussian
- \( f(x) \) approximated via a GP with EQ covariance
Bayesian Quadrature (BQ)

Evaluate integral of (expensive) black-box functions

\[ Z = \int p(x)f(x)\,dx \]

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

\[ \Rightarrow \text{posterior } Z \text{ can be computed analytically} \]
Bayesian Quadrature (BQ)

Evaluate integral of (expensive) black-box functions

\[ Z = \int p(x)f(x) \, dx \]

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

\[ \Rightarrow \text{posterior } Z \text{ can be computed analytically} \]

Bayesian Quadrature (BQ)

Evaluate integral of (expensive) black-box functions

\[
Z = \int p(x)f(x)dx
\]

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

$\implies$ posterior $Z$ can be computed analytically

BQ for Bayesian inference (previous work)

Evaluate marginal likelihood of (expensive) black-box functions
BQ for Bayesian inference (previous work)

- 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

- 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

- 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, \textit{NIPS} (2003)
- Doubly-Bayesian quadrature (BBQ), Osborne et al., \textit{NIPS} (2012)
- Warped seq. active Bayesian integration (WSABI), Gunter et al., \textit{NIPS} (2014)

Active sampling

- Minimize expected variance of integral $Z$
- \textit{Uncertainty sampling}: Maximize variance of integrand $p(x)f(x)$
Putting things together

Variational inference:

\[ \argmax_{\phi} \mathbb{E}_{q_{\phi}(x)} \left[ \log p(D|x) p(x) \right] + H[q_{\phi}(x)] \]

Bayesian quadrature:

\[ Z = \int q(x) f(x) \, dx \]

VI + BQ ⇒ VBMC
Putting things together

- Variational inference:

\[ q_\phi(x) = \arg\max_\phi \text{ELBO}(\phi) \]

\[ = \arg\max_\phi \left\{ \int q_\phi(x) \log [p(D|x)p(x)] \, dx + \mathcal{H}[q_\phi(x)] \right\} \]
Putting things together

- Variational inference:

\[ q_\phi(x) = \text{argmax}_\phi \text{ELBO}(\phi) \]
\[ = \text{argmax}_\phi \left\{ \int q_\phi(x) \log [p(D|x)p(x)] dx + H[q_\phi(x)] \right\} \]

- Bayesian quadrature:

\[ Z = \int q(x)f(x)dx \]
Putting things together

- Variational inference:

  \[ q_\phi(x) = \operatorname{argmax}_\phi \text{ELBO}(\phi) \]

  \[ = \operatorname{argmax}_\phi \left\{ \int q_\phi(x) \log [p(D|x)p(x)] \, dx + \mathcal{H}[q_\phi(x)] \right\} \]

- Bayesian quadrature:

  \[ Z = \int q(x) f(x) \, dx \]
Putting things together

- Variational inference:

\[ q_\phi(x) = \arg\max_{\phi} \text{ELBO}(\phi) \]

\[ = \arg\max_{\phi} \left\{ \int q_\phi(x) \log [p(D|x)p(x)] \, dx + \mathcal{H}[q_\phi(x)] \right\} \]

- Bayesian quadrature:

\[ Z = \int q(x)f(x) \, dx \]

\[ \text{VI + BQ} \Rightarrow \text{VBMC} \]
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 $f = \log p(D|x_{\text{new}})p(x_{\text{new}})$
2. train GP model of the log joint $f$
3. update variational posterior $q_{\phi_t}$ by optimizing the ELBO

Loop until reaching termination criterion

Acerbi, NeurIPS (2018)
VBMC demo

Target density

Iteration 0 (initial design)

Model evidence

-2.27

-4

Iterations

ELBO
LML

Luigi Acerbi

VBMC

Nov 26, 2018 21 / 40
VBMC demo

Target density

Iteration 2 (warm-up)

Model evidence

- Target density
- Iteration 2 (warm-up)
- Model evidence

Luigi Acerbi
VBMC demo

Target density

Iteration 3 (warm-up)

Model evidence

- Target density
- Iteration 3 (warm-up)
- Model evidence

Luigi Acerbi
**VBMC demo**

**Target density**

**Iteration 4 (warm-up)**

**Model evidence**

- **Target density**
  - Contour plot showing the target density distribution.

- **Iteration 4 (warm-up)**
  - Scatter plot with iterations marked.

- **Model evidence**
  - Graph showing model evidence over iterations:
    - **ELBO** (red line)
    - **LML** (black line)
  - Iterations range from 1 to 4:
    - At iteration 4, the model evidence values are:
      - ELBO: -2.27
      - LML: -1
Iteration 5 (warm-up)

Target density

Model evidence

-2.27

ELBO
LML

Luigi Acerbi
VBMC
Nov 26, 2018 21 / 40
VBMC demo

Target density

Iteration 6 (end of warm-up)

Model evidence

Target density

Iteration 6 (end of warm-up)

Model evidence

-2.27
VBMC demo

Target density

Iteration 11

Model evidence

- Target density
- Iteration 11
- Model evidence

Luigi Acerbi
VBMC
Nov 26, 2018
Target density

Iteration 12

Model evidence

-4
-2.27
-1

Model evidence

ELBO
LML
VBMC demo

Target density

Iteration 14

Model evidence

-2.27

x 1

x 2

ELBO

LML

-4

-1

1

5

10

14

Iterations

x 1

x 2

-4

-2.27

1

14

Iterations

ELBO

LML
VBMC demo

Target density

Iteration 15

Model evidence

|x1| x2| x1| x2|

-4 -2.27 -1

Model evidence

ELBO
LML

Luigi Acerbi
Variational posterior

\[ q_\phi(x) = \sum_{k=1}^{K} w_k \mathcal{N}(x; \mu_k, \sigma_k^2 \Sigma), \quad \Sigma \equiv \text{diag}[\lambda_1^{(2)}, \ldots, \lambda_D^{(2)}] \]
Variational posterior

\[
q_\phi(x) = \sum_{k=1}^{K} w_k \mathcal{N}(x; \mu_k, \sigma_k^2 \Sigma), \quad \Sigma \equiv \text{diag}[\lambda^{(1)}^2, \ldots, \lambda^{(D)}^2]
\]

- \(x \in \mathbb{R}^D\)
- \(\phi \equiv (w_1, \ldots, w_K, \mu_1, \ldots, \mu_K, \sigma_1, \ldots, \sigma_K, \lambda)\)
- \(K(D + 2) + D\) parameters
- \(K\) is changed adaptively each iteration
Gaussian process representation

\[ f(x) = \log p(D|x)p(x) \]
Gaussian process representation

\[ f(x) = \log p(D|x)p(x) \]

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

\[ f(x) = \log p(D|x)p(x) \]

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

\[ m_{\text{NQ}}(x) = m_0 - \frac{1}{2} \sum_{i=1}^{D} \frac{(x(i) - x_{m}(i))^2}{\omega(i)^2}, \]
Gaussian process representation

\[ f(x) = \log p(D|x)p(x) \]

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

\[ m_{NQ}(x) = m_0 - \frac{1}{2} \sum_{i=1}^{D} \frac{\left(x^{(i)} - x_m^{(i)}\right)^2}{\omega(i)^2} \]

Sample over GP hyperparameters (later optimize)
Variational optimization

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

\[
\text{ELBO}(\phi, f) = \int q_\phi(x)f(x)dx + H[q_\phi(x)]
\]

- Expected log joint and gradient are analytical
Variational optimization

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

- 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

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

- 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: \( x_{\text{next}} = \arg \max_x a(x) \)
Active sampling

Optimize acquisition function: \( x_{\text{next}} = \arg \max_x a(x) \)

**Goal:** Evaluate \( \mathbb{E}_\phi [f] = \int q_\phi(x)f(x)dx \)
Active sampling

Optimize acquisition function: \( x_{\text{next}} = \arg \max_x a(x) \)

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

\( \Rightarrow \) ‘Vanilla’ uncertainty sampling: \( a_{\text{us}}(x) = V(x)q_\phi(x)^2 \)
Active sampling

Optimize acquisition function: \( x_{\text{next}} = \arg \max_x a(x) \)

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

\[ \implies \text{‘Vanilla’ uncertainty sampling:} \quad a_{\text{us}}(x) = V(x)q_\phi(x)^2 \]

Goal: Evaluate \( \mathbb{E}_{\phi_1} [f], \mathbb{E}_{\phi_2} [f], \ldots, \mathbb{E}_{\phi_T} [f] \)
Active sampling

Optimize acquisition function: \( x_{\text{next}} = \arg \max_x a(x) \)

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

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

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

\( \Rightarrow \) Prospective uncertainty sampling: \( a_{\text{pro}}(x) = V(x)q_\phi(x) \exp(\bar{f}(x)) \)
Algorithmic details

Evidence Lower Confidence Bound (ELCBO)

$$\text{ELCBO} (\phi, f) = \text{ELBO} (\phi, f) - \beta \cdot \text{LCB} \cdot \text{SD} \left[ E \phi [f] \right]$$

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 $n_{\text{stable}}$ iterations
Algorithmic details

Evidence Lower Confidence Bound (ELCBO)

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

**Evidence Lower Confidence Bound (ELCBO)**

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

- 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)**

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

- **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)

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

- 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 \( n_{\text{stable}} \) 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 \[ D \in \{2, 4, 6, 8, 10\} \]
Synthetic target densities

Three families: *Lumpy, Student, Cigar*  \[ D \in \{2, 4, 6, 8, 10\} \]
Synthetic target densities: Results

- Lumpy
- Student
- Cigar

2D 6D 10D

Median LML error

Function evaluations

- smc
- ais
- bmc
- wsabi-L
- wsabi-M
- bbq
- agp
- bape
- vbmc-U
- vbmc-P

Luigi Acerbi
VBMC
Nov 26, 2018
Synthetic target densities: Results

Lumpy

Student

Cigar

Median LML error

Function evaluations

Luigi Acerbi

VBMC

Nov 26, 2018
Synthetic target densities: Results

Lumpy

Student

Cigar

Function evaluations

Median LML error

100 200 400

10^{-4} 10^{-2} 1 10^{2} 10^{4}

200 400 600

Function evaluations

Median LML error

10^{-4} 10^{-2} 1 10^{2} 10^{4}

Function evaluations

Median LML error

10^{-4} 10^{-2} 1 10^{2} 10^{4}

Function evaluations
Synthetic target densities: Results

Lumpy

Student

Cigar

Median gsKL
Neuronal model: Results

Two datasets: V1, V2

\[ D = 7 \]

---

**V1**

- Neuronal model

**V2**

- Neuronal model

Function evaluations: 10^{-2}, 1, 10^2, 10^4

Median LML error: 200 400
Neuronal model: Results

Two datasets: V1, V2  \( D = 7 \)

Neuronal model

![Graphs showing median log-likelihood (gsKL) vs. function evaluations for V1 and V2 datasets. The graphs illustrate the performance of different algorithms, including smc, ais, bmc, wsabi-L, wsabi-M, bbq, agp, bape, vbmc-U, and vbmc-P, across different numbers of function evaluations.](image-url)
Neuronal model: VBMC

VBMC (iteration 52)
Computational cost

Neuronal model (V1)

Median algorithmic cost per function evaluation (s)

Function evaluations

Median algorithmic cost per function evaluation (s)

Neuronal model (V1)

Function evaluations

Median algorithmic cost per function evaluation (s)

Neuronal model (V1)

Function evaluations

Median algorithmic cost per function evaluation (s)

Neuronal model (V1)

Function evaluations

Median algorithmic cost per function evaluation (s)

Neuronal model (V1)

Function evaluations

Median algorithmic cost per function evaluation (s)

Neuronal model (V1)

Function evaluations

Median algorithmic cost per function evaluation (s)

Neuronal model (V1)

Function evaluations

Median algorithmic cost per function evaluation (s)

Neuronal model (V1)

Function evaluations

Median algorithmic cost per function evaluation (s)

Neuronal model (V1)

Function evaluations

Median algorithmic cost per function evaluation (s)

Neuronal model (V1)

Function evaluations

Median algorithmic cost per function evaluation (s)

Neuronal model (V1)

Function evaluations

Median algorithmic cost per function evaluation (s)

Neuronal model (V1)

Function evaluations

Median algorithmic cost per function evaluation (s)

Neuronal model (V1)

Function evaluations

Median algorithmic cost per function evaluation (s)
What’s the secret sauce?
What’s the secret sauce?

Other quadrature methods: (BMC, BBQ, WSABI)

\[ Z = \int p(x)p(D|x)dx \]
What’s the secret sauce?

Other quadrature methods: (BMC, BBQ, WSABI)

\[ Z = \int p(x)p(D|x)dx \]

VBMC:

\[ I_k = \int q_k(x) \log [p(x)p(D|x)] \, dx \]
What’s the secret sauce?

Other quadrature methods: (BMC, BBQ, WSABI)

$$Z = \int p(x)p(D|x)dx$$

VBMC:

$$\mathcal{I}_k = \int q_k(x) \log [p(x)p(D|x)] dx$$

- GP representation
What’s the secret sauce?

Other quadrature methods: (BMC, BBQ, WSABI)

\[ Z = \int p(x)p(D|x)dx \]

VBMC:

\[ I_k = \int q_k(x) \log [p(x)p(D|x)] dx \]

- 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**

Variational Bayesian Monte Carlo (VBMC) algorithm for posterior and model inference in MATLAB

- bayesian-inference
- variational-inference
- gaussian-processes
- data-analysis
- machine-learning
- matlab

- 344 commits
- 1 branch
- 0 releases
- 1 contributor
- GPL-3.0

**lacerbi / bads**

Bayesian Adaptive Direct Search (BADS) optimization algorithm for model fitting in MATLAB

- optimization-algorithms
- baysian-optimization
- log-likelihood
- noiseless-functions
- noisy-functions
- matlab

- 156 commits
- 2 branches
- 6 releases
- 1 contributor
- GPL-3.0

Final slide

- VBMC toolbox at: github.com/lacerbi/vbmc
- BADS toolbox at: github.com/lacerbi/bads
Final slide

- 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 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 WSABLI-L on VBMC samples

Neuronal model

Function evaluations

Median LML error

V1

V2

vbmc-P
vbmc-control
wsabi-L