

# Introduction to Bayesian Inference for Statistical Model Fitting (DRAFT)

Luigi Acerbi

Department of Computer Science  
University of Helsinki  
Finnish Center for Artificial Intelligence FCAI



**FCAI** Finnish  
Center for  
Artificial  
Intelligence



BAMB! Summer School – Day 2  
September 2022

- 1 Introduction and motivation
  - Bayes rule
  - Bayesian inference for model fitting
- 2 Computing the posterior distribution
  - Computing the posterior “by hand”
  - Inference algorithms
  - The prior
- 3 Making use of a Bayesian posterior
  - Visualizing the posterior
  - Posterior prediction

- 1 Introduction and motivation
  - Bayes rule
  - Bayesian inference for model fitting
- 2 Computing the posterior distribution
  - Computing the posterior “by hand”
  - Inference algorithms
  - The prior
- 3 Making use of a Bayesian posterior
  - Visualizing the posterior
  - Posterior prediction

# What is Bayesian inference?

# What is Bayesian inference?



My rule.

$$p(\theta|\text{data}) = \frac{p(\text{data}|\theta)p(\theta)}{p(\text{data})}$$

# What is Bayesian inference?



My rule.

$$\overbrace{p(\theta|\text{data})}^{\text{posterior}} = \frac{\overbrace{p(\text{data}|\theta)}^{\text{likelihood}} \overbrace{p(\theta)}^{\text{prior}}}{\underbrace{p(\text{data})}_{\text{evidence}}}$$

# What is Bayesian inference?



My rule.

$$\overbrace{p(\theta|\text{data})}^{\text{posterior}} = \frac{\overbrace{p(\text{data}|\theta)}^{\text{likelihood}} \overbrace{p(\theta)}^{\text{prior}}}{\underbrace{p(\text{data})}_{\text{evidence}}}$$

$$p(\text{data}) = \int p(\text{data}|\theta)p(\theta)d\theta$$

# Where does Bayes rule come from?



# Where does Bayes rule come from?

From me.



# Where does Bayes rule come from?



From me.

Really, just basic rules of probability:

# Where does Bayes rule come from?



From me.

Really, just basic rules of probability:

①  $p(\theta, \text{data}) = p(\theta|\text{data})p(\text{data})$

# Where does Bayes rule come from?



From me.

Really, just basic rules of probability:

- 1  $p(\theta, \text{data}) = p(\theta|\text{data})p(\text{data})$
- 2  $p(\theta, \text{data}) = p(\text{data}|\theta)p(\theta)$

# Where does Bayes rule come from?



From me.

Really, just basic rules of probability:

- 1  $p(\theta, \text{data}) = p(\theta|\text{data})p(\text{data})$
- 2  $p(\theta, \text{data}) = p(\text{data}|\theta)p(\theta)$
- 3  $p(\theta|\text{data})p(\text{data}) = p(\text{data}|\theta)p(\theta)$

# Where does Bayes rule come from?



From me.

Really, just basic rules of probability:

- ①  $p(\theta, \text{data}) = p(\theta|\text{data})p(\text{data})$
- ②  $p(\theta, \text{data}) = p(\text{data}|\theta)p(\theta)$
- ③  $p(\theta|\text{data})p(\text{data}) = p(\text{data}|\theta)p(\theta)$
- ④  $p(\theta|\text{data}) = \frac{p(\text{data}|\theta)p(\theta)}{p(\text{data})}$

# What's new in Bayesian inference for model fitting?

The output of Bayesian inference is a **probability distribution** (posterior) over model parameters:

$$p(\boldsymbol{\theta}|\text{data})$$

Before, we only had a single best **point estimate**  $\boldsymbol{\theta}_\star$ .

# What's new in Bayesian inference for model fitting?

The output of Bayesian inference is a **probability distribution** (posterior) over model parameters:

$$p(\boldsymbol{\theta}|\text{data})$$

Before, we only had a single best **point estimate**  $\boldsymbol{\theta}_\star$ .

Questions:

- 1 How do we compute  $p(\boldsymbol{\theta}|\text{data})$ ?
- 2 What do we do once we have  $p(\boldsymbol{\theta}|\text{data})$ ?
- 3 Why should we bother?



# Why Bayesian inference?

$$\overbrace{p(\boldsymbol{\theta}|\text{data})}^{\text{posterior}} = \frac{\overbrace{p(\text{data}|\boldsymbol{\theta})}^{\text{likelihood}} \overbrace{p(\boldsymbol{\theta})}^{\text{prior}}}{\underbrace{p(\text{data})}_{\text{evidence}}}$$

$$p(\text{data}) = \int p(\text{data}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}$$

# Why Bayesian inference?

$$\overbrace{p(\theta|\text{data})}^{\text{posterior}} = \frac{\overbrace{p(\text{data}|\theta)}^{\text{likelihood}} \overbrace{p(\theta)}^{\text{prior}}}{\underbrace{p(\text{data})}_{\text{evidence}}}$$

$$p(\text{data}) = \int p(\text{data}|\theta)p(\theta)d\theta$$

- Uncertainty quantification
- Optimal experiment design
- Robustness
- Interpretability

# Why Bayesian inference?

$$\overbrace{p(\boldsymbol{\theta}|\text{data})}^{\text{posterior}} = \frac{\overbrace{p(\text{data}|\boldsymbol{\theta})}^{\text{likelihood}} \overbrace{p(\boldsymbol{\theta})}^{\text{prior}}}{\underbrace{p(\text{data})}_{\text{evidence}}}$$

$$p(\text{data}) = \int p(\text{data}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}$$

- Uncertainty quantification
- Optimal experiment design
- Robustness
- Interpretability
- Hyperparameter tuning
- Model selection

# Why Bayesian inference?

$$\underbrace{p(\theta|\text{data})}_{\text{posterior}} = \frac{\overbrace{p(\text{data}|\theta)}^{\text{likelihood}} \overbrace{p(\theta)}^{\text{prior}}}{\underbrace{p(\text{data})}_{\text{evidence}}}$$

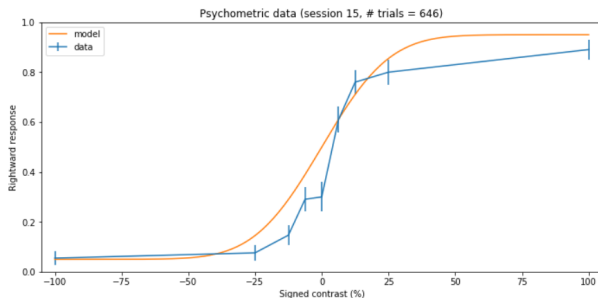
$$p(\text{data}) = \int p(\text{data}|\theta)p(\theta)d\theta$$

- Uncertainty quantification
- Optimal experiment design
- Robustness
- Interpretability
- Better predictions
- Hyperparameter tuning
- Model selection

- 1 Introduction and motivation
  - Bayes rule
  - Bayesian inference for model fitting
- 2 Computing the posterior distribution
  - Computing the posterior “by hand”
  - Inference algorithms
  - The prior
- 3 Making use of a Bayesian posterior
  - Visualizing the posterior
  - Posterior prediction

# Data and model

- Same data from before (IBL mouse behavioral data)
- Same model as before (psychometric function model)
  - ▶ Parameters  $\theta = (\mu, \sigma, \lambda, \gamma)$



# Let's just apply Bayes rule!

- We consider a 1-D posterior (one free parameter,  $\sigma$ )
- We fix  $\mu, \lambda, \gamma$  to some values  $\mu_*, \lambda_*, \gamma_*$
- We assume a uniform prior  $p(\sigma)$  for  $\sigma \in [1, 100]$
- We compute

$$p(\sigma|\mu_*, \lambda_*, \gamma_*, \text{data}) = \frac{p(\text{data}|\mu_*, \sigma, \lambda_*, \gamma_*)p(\sigma)}{Z}$$

- The normalization is  $Z = \int p(\text{data}|\mu_*, \sigma, \lambda_*, \gamma_*)p(\sigma)d\sigma$

Let's do this!

# Bayesian inference solved?

- Not really – a grid only works in low dimension ( $D \sim 1 - 4$ )
- Curse of dimensionality:  $N$  points per dimension  $\Rightarrow N^D$  points
- We need **inference algorithms**!



# Inference algorithms

- A general-purpose inference algorithm
  - ▶ takes as input an inference problem (likelihood, prior, ...)
  - ▶ returns an **approximate posterior**
- Abstractly, similar to optimization...
  - ▶ take as input an optimization problem (target function)
  - ▶ return the optimum
- ... in practice, way more complex algorithms
  - ▶ Inference is **harder**

# Main families of inference algorithms

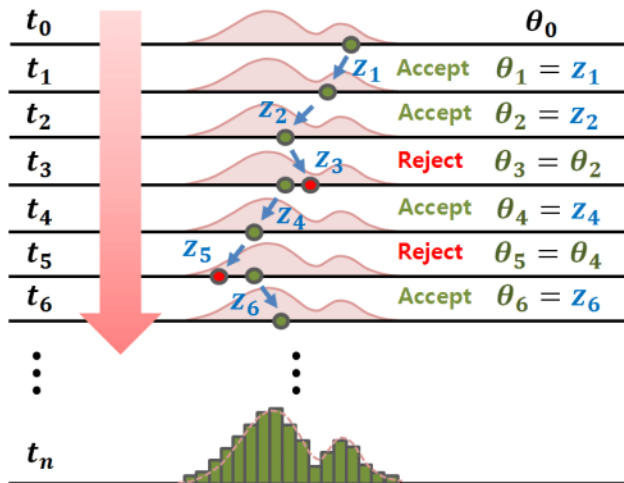
- ① Markov Chain Monte Carlo (MCMC)
- ② Variational inference

(there are others)

# Markov Chain Monte Carlo (MCMC)

- Generates a random sequence  $\theta_0, \theta_1, \dots$  (a Markov chain)
- Various rules for drawing  $\theta_{n+1}|\theta_n$  depending on the algorithm
  - ▶ These will generally depend on  $p(\theta_n, \text{data})$ ,  $p(\theta_{n+1}, \text{data})$
- **Output:** A set of samples  $\theta_0, \dots, \theta_N$
- **If all goes well,**  $\theta_0, \dots, \theta_N \sim p(\theta|\text{data})$ 
  - ▶ In practice, lot of tweaking to ensure **convergence** of the Markov chain
  - ▶ State-of-the-art MCMC methods are (to a degree) **self-tuning**
  - ▶ Still a lot of tweaking involved

# Example MCMC algorithm: Metropolis-Hastings



Source: Jin et al. (2019)

# Variational inference

- Approximate  $p(\theta|\text{data})$  with  $q_\phi(\theta)$

# Variational inference

- Approximate  $p(\theta|\text{data})$  with  $q_\phi(\theta)$
- Minimize Kullback-Leibler divergence between  $q$  and  $p$

# Variational inference

- Approximate  $p(\boldsymbol{\theta}|\text{data})$  with  $q_{\phi}(\boldsymbol{\theta})$
- Minimize Kullback-Leibler divergence between  $q$  and  $p$

Obtains

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

# Variational inference

- Approximate  $p(\boldsymbol{\theta}|\text{data})$  with  $q_{\phi}(\boldsymbol{\theta})$
- Minimize Kullback-Leibler divergence between  $q$  and  $p$

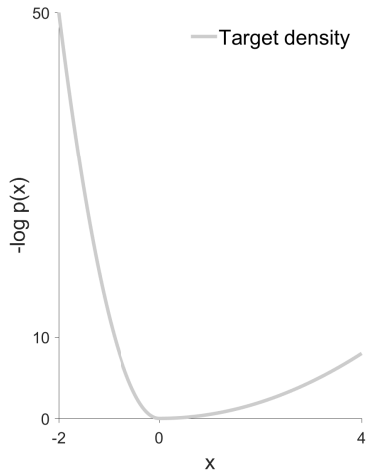
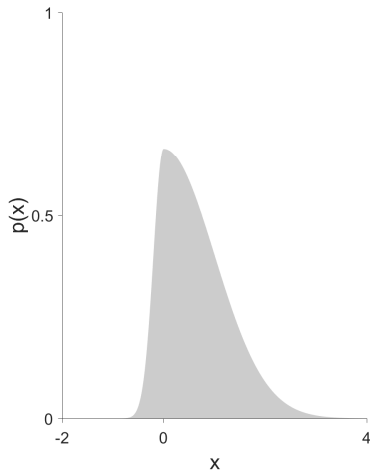
Obtains

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

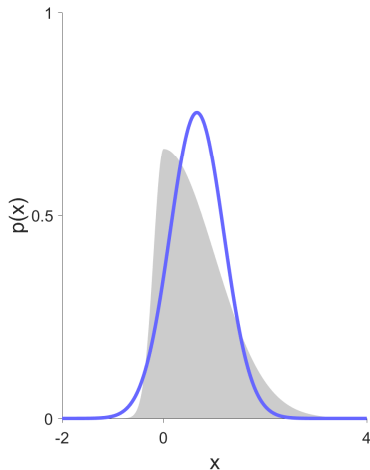
VI casts Bayesian inference into optimization + integration



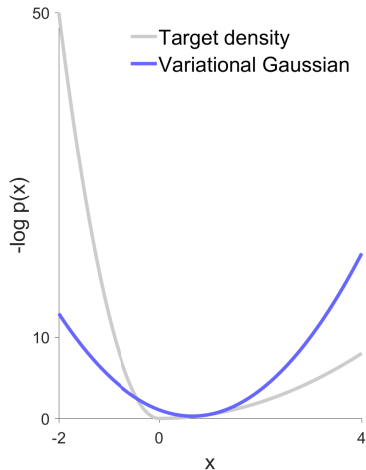
# Variational inference: example



# Variational inference: example

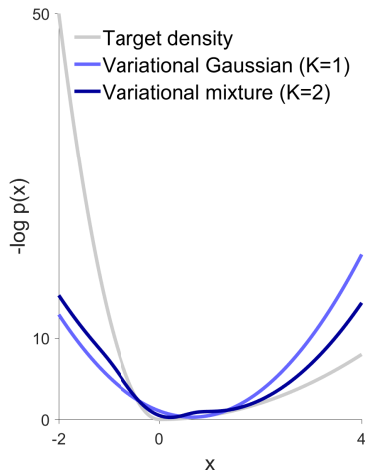
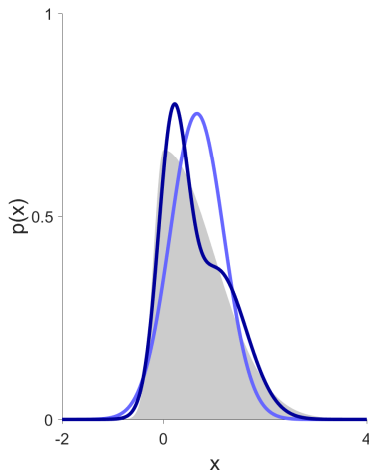


$$q_{\phi}(x) = \mathcal{N}(x, \mu, \sigma^2)$$



$$\phi = (\mu, \sigma^2)$$

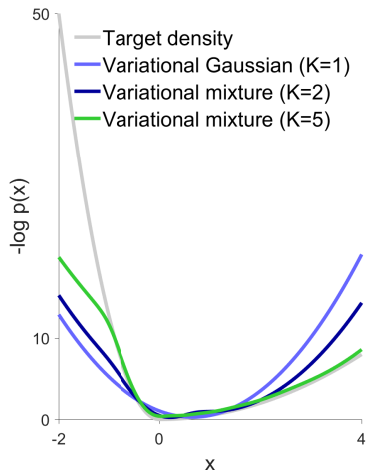
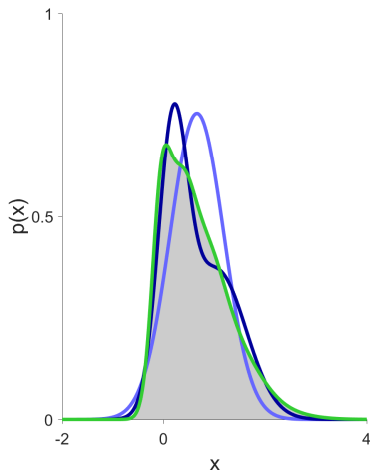
# Variational inference: example



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

$$\phi = (w_k, \mu_k, \sigma_k^2)_{k=1}^K$$

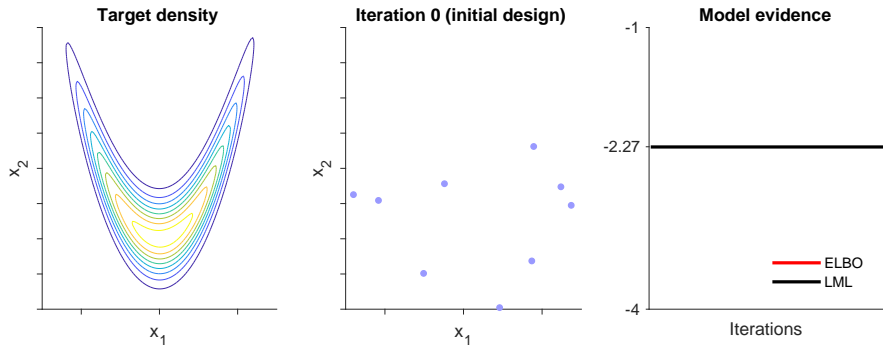
# Variational inference: example



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

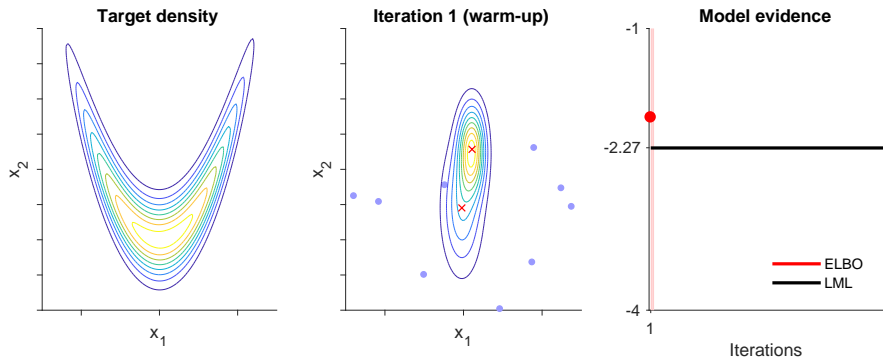
$$\phi = (w_k, \mu_k, \sigma_k^2)_{k=1}^K$$

# Variational Bayesian Monte Carlo (VBMC)



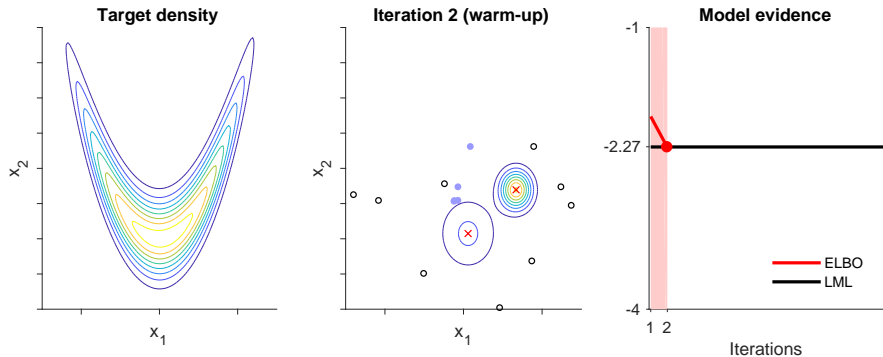
Acerbi, *NeurIPS* (2018; 2020)

# Variational Bayesian Monte Carlo (VBMC)



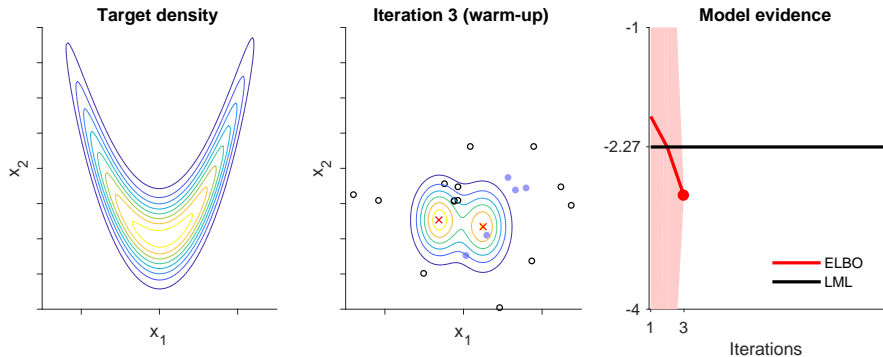
Acerbi, *NeurIPS* (2018; 2020)

# Variational Bayesian Monte Carlo (VBMC)



Acerbi, *NeurIPS* (2018; 2020)

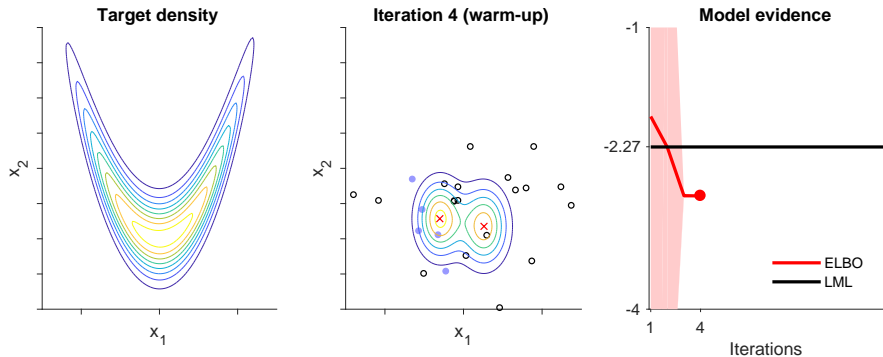
# Variational Bayesian Monte Carlo (VBMC)



Acerbi, *NeurIPS* (2018; 2020)

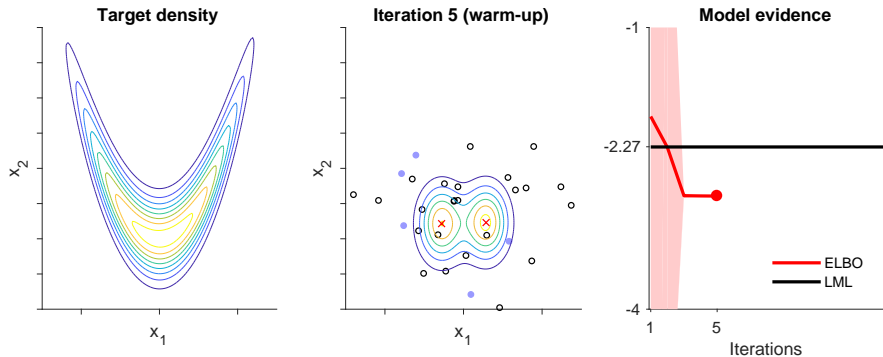


# Variational Bayesian Monte Carlo (VBMC)



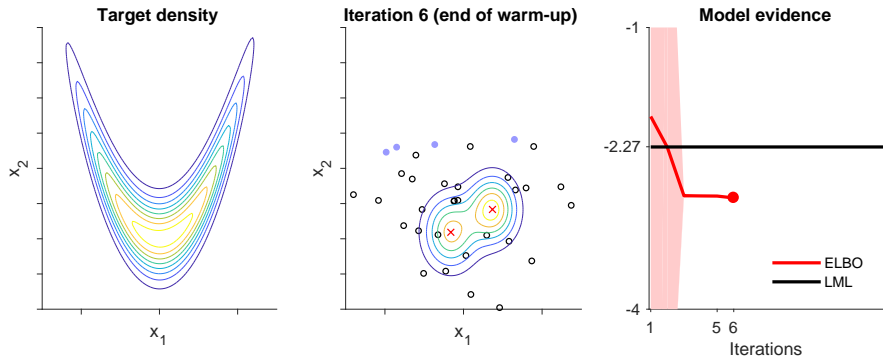
Acerbi, *NeurIPS* (2018; 2020)

# Variational Bayesian Monte Carlo (VBMC)



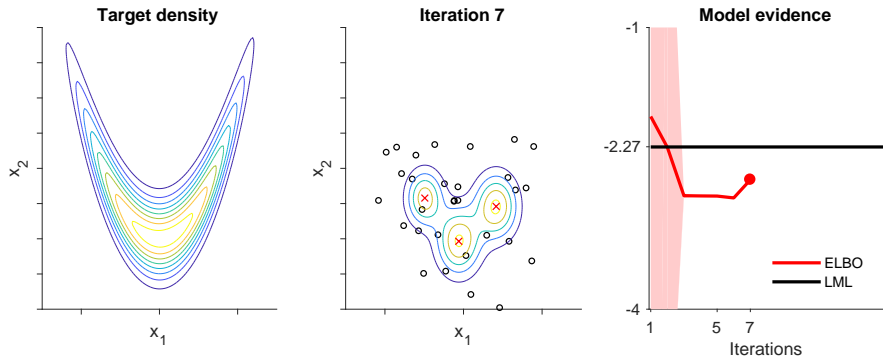
Acerbi, *NeurIPS* (2018; 2020)

# Variational Bayesian Monte Carlo (VBMC)



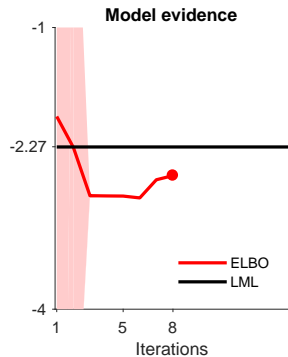
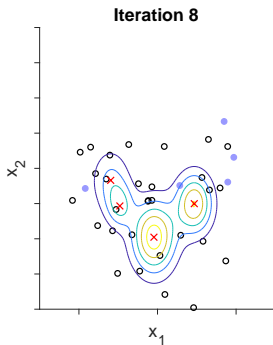
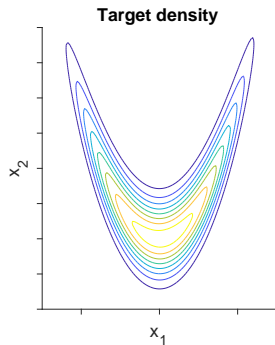
Acerbi, *NeurIPS* (2018; 2020)

# Variational Bayesian Monte Carlo (VBMC)



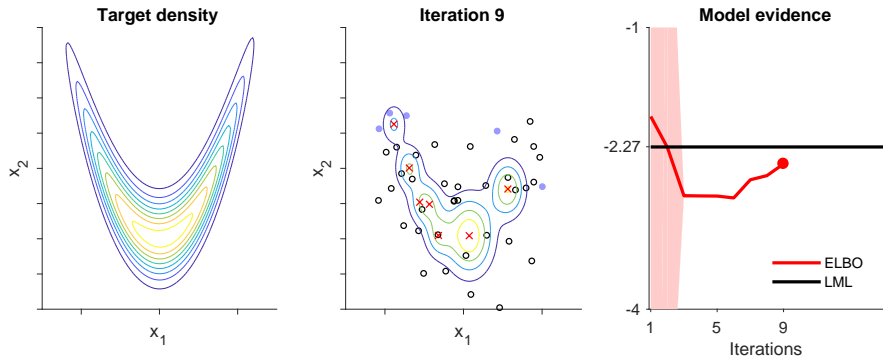
Acerbi, *NeurIPS* (2018; 2020)

# Variational Bayesian Monte Carlo (VBMC)



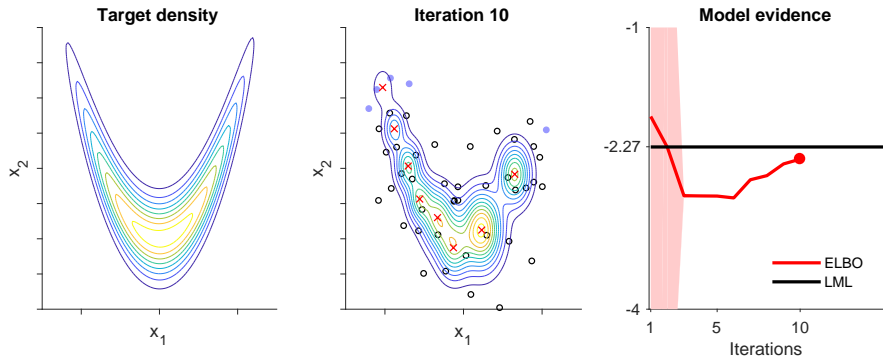
Acerbi, *NeurIPS* (2018; 2020)

# Variational Bayesian Monte Carlo (VBMC)



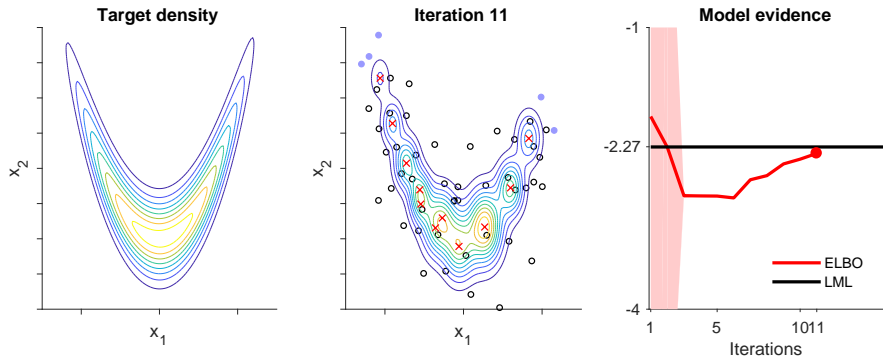
Acerbi, *NeurIPS* (2018; 2020)

# Variational Bayesian Monte Carlo (VBMC)



Acerbi, *NeurIPS* (2018; 2020)

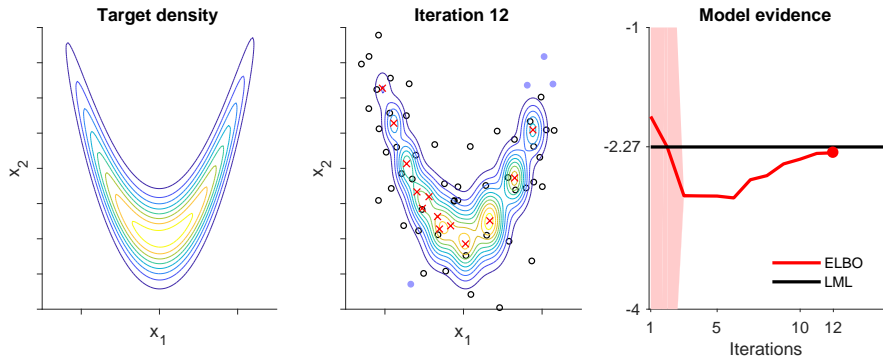
# Variational Bayesian Monte Carlo (VBMC)



Acerbi, *NeurIPS* (2018; 2020)

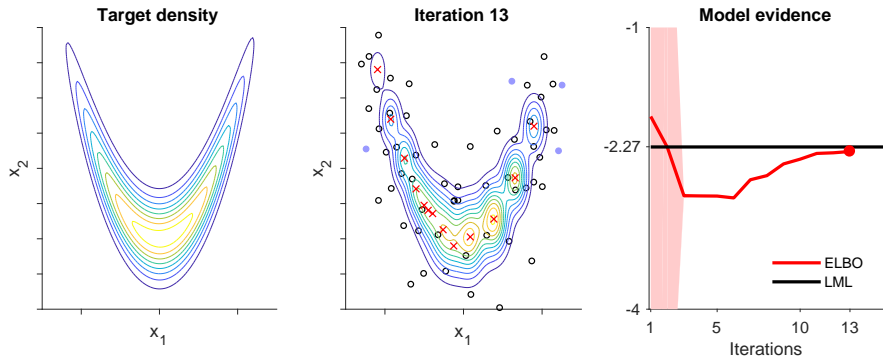


# Variational Bayesian Monte Carlo (VBMC)



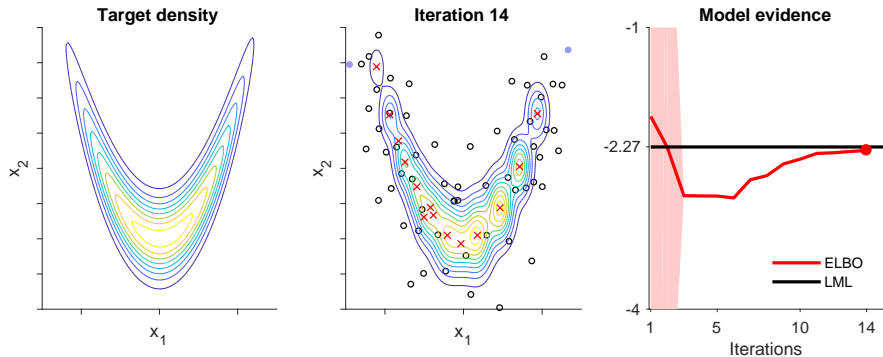
Acerbi, *NeurIPS* (2018; 2020)

# Variational Bayesian Monte Carlo (VBMC)



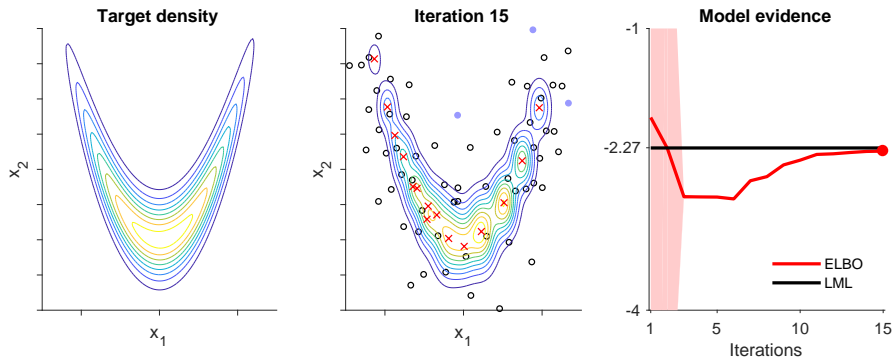
Acerbi, *NeurIPS* (2018; 2020)

# Variational Bayesian Monte Carlo (VBMC)



Acerbi, *NeurIPS* (2018; 2020)

# Variational Bayesian Monte Carlo (VBMC)



Acerbi, *NeurIPS* (2018; 2020)

# Pick your prior

- In Bayesian inference you need a **prior** over parameters,  $p(\theta)$
- Common choice: independent priors  $p(\theta) = \prod_{d=1}^D p(\theta_d)$ 
  - ▶ Choose the prior  $p(\theta_d)$  for each parameter
  - ▶ Independent prior does not mean that the posterior is independent!
- Remember that the prior is a probability distribution  $\int p(\theta) d\theta = 1$
- Okay, but how do I pick a prior for each parameter?

## Example priors: uniform box

- Bounded parameter
- Uniform in the full range
- **Pros:** Easy to define and to justify
- **Cons:** Non-informative

## Example priors: tent

- Bounded parameter
- Uniform in a range, then falls off
- Uses the hard/plausible bounds defined previously
- **Pros:** Still easy to define, “weakly” informative
- **Cons:** Need some thought to define the plausible range

## Example priors: smoothed tent

- Bounded parameter
- Just like tent prior but with smooth edges
- **Pros:** Better numerical properties than tent prior
- **Cons:** More complex to implement (can use provided functions)



OK so we have a posterior what now

...

# Predictions

...

# Final slide

- Contact me at `luigi.acerbi@helsinki.fi`
- Optimization demos: `github.com/lacerbi/optimviz`

## **MATLAB toolboxes:**

- BADS available at `github.com/lacerbi/bads`

# Final slide

- Contact me at `luigi.acerbi@helsinki.fi`
- Optimization demos: `github.com/lacerbi/optimviz`

## **MATLAB toolboxes:**

- BADS available at `github.com/lacerbi/bads`

Thanks!

(Time for questions?)