Introduction to Bayesian Inference for Statistical Model Fitting (DRAFT)

Luigi Acerbi

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

BAMB! Summer School – Day 2
September 2022

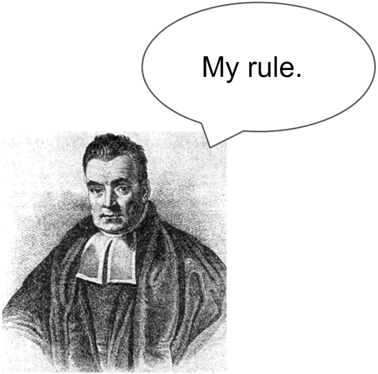
Learning objectives

By the end of this lecture/tutorial, we will:

- ...

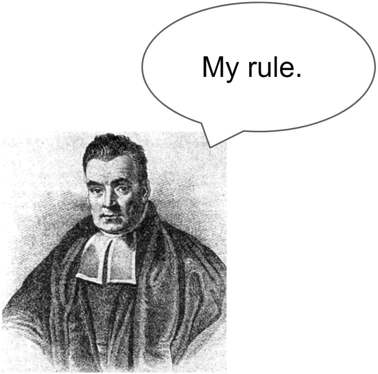
What is Bayesian inference?

What is Bayesian inference?



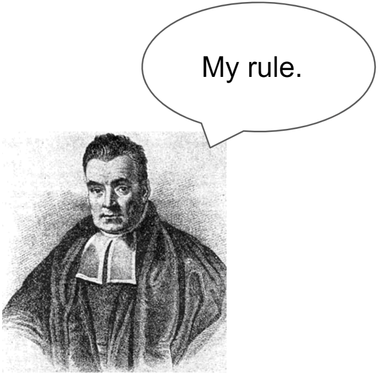
$$p(\boldsymbol{\theta}|\text{data}) = \frac{p(\text{data}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\text{data})}$$

What is Bayesian inference?



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

What is Bayesian inference?

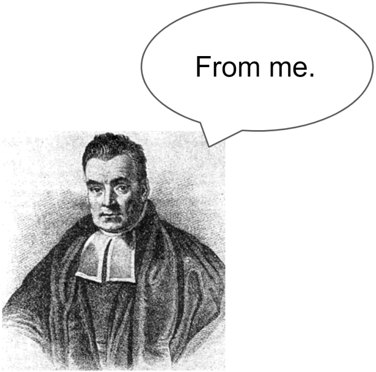


$$\overbrace{p(\boldsymbol{\theta}|\text{data})}^{\text{posterior}} = \frac{\overbrace{p(\text{data}|\boldsymbol{\theta})}^{\text{likelihood}}\,\overbrace{p(\boldsymbol{\theta})}^{\text{prior}}}{\underbrace{p(\text{data})}_{\text{evidence}}}$$

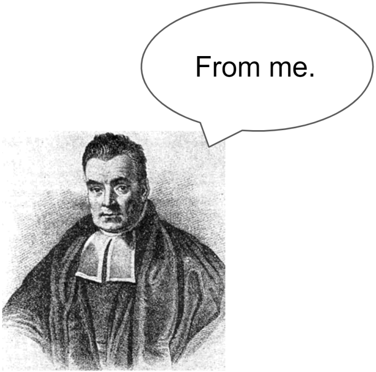
$$p(\text{data}) = \int p(\text{data}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}$$

Where does Bayes rule come from?

Where does Bayes rule come from?

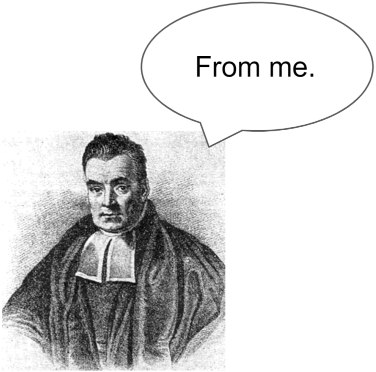


Where does Bayes rule come from?



Really, just basic rules of probability:

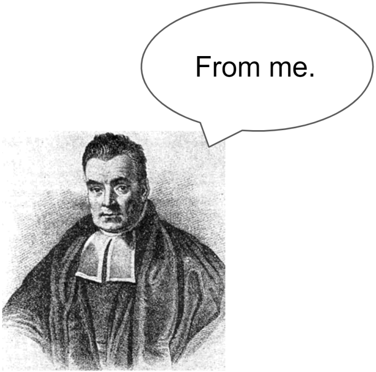
Where does Bayes rule come from?



Really, just basic rules of probability:

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

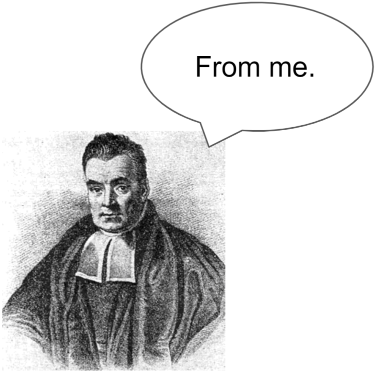
Where does Bayes rule come from?



Really, just basic rules of probability:

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

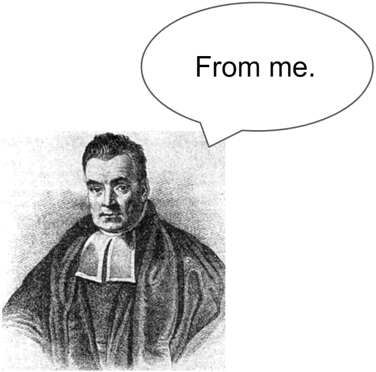
Where does Bayes rule come from?



Really, just basic rules of probability:

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

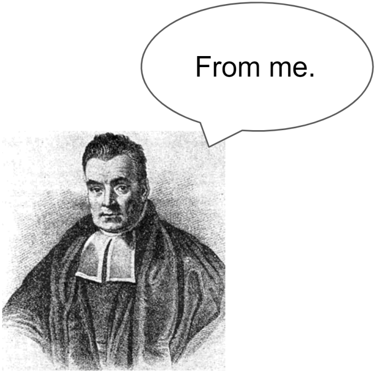
Where does Bayes rule come from?



Really, just basic rules of probability:

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

Where does Bayes rule come from?



Really, just basic rules of probability:

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

Bayesian probability

- We are treating both data and $\boldsymbol{\theta}$ as random variables.
- Probability as degree of belief.

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?

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?

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

Why Bayesian inference?

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

- 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?

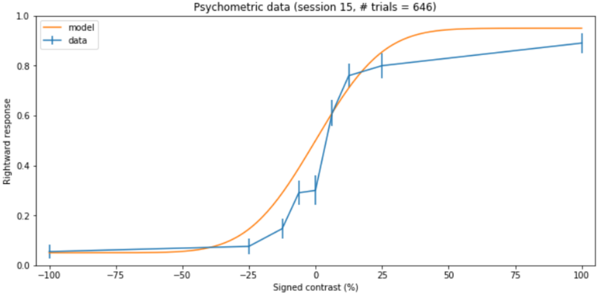
$$\overbrace{\color{red}p(\boldsymbol{\theta}|\text{data})}^{\color{red}\text{posterior}} = \frac{\overbrace{p(\text{data}|\boldsymbol{\theta})}^{\text{likelihood}}\,\overbrace{p(\boldsymbol{\theta})}^{\text{prior}}}{\underbrace{\color{red}p(\text{data})}_{\color{red}\text{evidence}}} \qquad p(\text{data}) = \int p(\text{data}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}$$

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

- Better predictions

Data and model

- Same data from before (IBL mouse behavioral data)
- Same model as before (psychometric function model)



Example: Let's apply Bayes rule

Example: Let's apply Bayes rule

- Model parameters $\boldsymbol{\theta} = (\mu, \sigma, \lambda, \gamma)$

Example: Let's apply Bayes rule

- Model parameters $\boldsymbol{\theta} = (\mu, \sigma, \lambda, \gamma)$
- For simplicity:
  - We fix μ, λ, γ to some values $\mu_\star, \lambda_\star, \gamma_\star$
  - One free parameter, σ

Example: Let's apply Bayes rule

- Model parameters $\boldsymbol{\theta} = (\mu, \sigma, \lambda, \gamma)$
- For simplicity:
  - We fix μ, λ, γ to some values $\mu_\star, \lambda_\star, \gamma_\star$
  - One free parameter, σ
- We compute

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

Example: Let's apply Bayes rule

- Model parameters $\boldsymbol{\theta} = (\mu, \sigma, \lambda, \gamma)$
- For simplicity:
  - We fix μ, λ, γ to some values $\mu_\star, \lambda_\star, \gamma_\star$
  - One free parameter, σ
- We compute

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

- We assume a uniform-box prior $p(\sigma)$ for $\sigma \in [1, 100]$

$$p(\sigma) = \begin{cases} \frac{1}{99} & \text{for } 1 \le \sigma \le 100 \\ 0 & \text{otherwise} \end{cases}$$

Example: Let's apply Bayes rule

- Model parameters $\boldsymbol{\theta} = (\mu, \sigma, \lambda, \gamma)$
- For simplicity:
  - We fix μ, λ, γ to some values $\mu_\star, \lambda_\star, \gamma_\star$
  - One free parameter, σ
- We compute

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

- We assume a uniform-box prior $p(\sigma)$ for $\sigma \in [1, 100]$

$$p(\sigma) = \begin{cases} \frac{1}{99} & \text{for } 1 \leq \sigma \leq 100 \\ 0 & \text{otherwise} \end{cases}$$

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

Hacking time I

Let's do Bayesian inference by hand!

Pick your prior

- In Bayesian inference you need a prior over parameters, $p(\boldsymbol{\theta})$

Pick your prior

- In Bayesian inference you need a prior over parameters, $p(\boldsymbol{\theta})$
- Common choice: independent priors $p(\boldsymbol{\theta}) = \prod_{d=1}^{D} p(\theta_d)$

Pick your prior

- In Bayesian inference you need a prior over parameters, $p(\boldsymbol{\theta})$
- Common choice: independent priors $p(\boldsymbol{\theta}) = \prod_{d=1}^{D} p(\theta_d)$
  - Choose the prior $p(\theta_d)$ for each parameter
  - Independent prior does not mean that the posterior is independent!

Pick your prior

- In Bayesian inference you need a prior over parameters, $p(\boldsymbol{\theta})$
- Common choice: independent priors $p(\boldsymbol{\theta}) = \prod_{d=1}^{D} p(\theta_d)$
  - 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 $\int p(\theta) d\theta = 1$

Pick your prior

- In Bayesian inference you need a prior over parameters, $p(\boldsymbol{\theta})$
- Common choice: independent priors $p(\boldsymbol{\theta}) = \prod_{d=1}^{D} p(\theta_d)$
  - 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 (if wide bounds)
- **Cons:** Non-informative

Example priors: tent/trapezoidal

- Bounded parameter
- Uniform in a range, then falls off
- Can use 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/trapezoidal

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

What about not-bounded parameters?

What about not-bounded parameters?

Unbounded $\theta \in (-\infty, \infty)$

- Gaussian distributions (with wide σ)
- Student's t distributions ($\nu = 3 - 7$)

What about not-bounded parameters?

Unbounded $\theta \in (-\infty, \infty)$

- Gaussian distributions (with wide σ)
- Student's t distributions ($\nu = 3 - 7$)

Half-bounded $\theta \in (0, \infty)$

- Gamma distributions
- Half-truncated Gaussians or t distributions

What about not-bounded parameters?

Unbounded $\theta \in (-\infty, \infty)$

- Gaussian distributions (with wide σ)
- Student's t distributions ($\nu = 3 - 7$)

Half-bounded $\theta \in (0, \infty)$

- Gamma distributions
- Half-truncated Gaussians or t distributions

Hot take:

- I generally recommend bounded parameters
- Half-bounded / unbounded parameters $\Rightarrow$ numerical issues

Hacking time II

Let's have a look at the priors.

Bayesian inference done?

Bayesian inference done?

- 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

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

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!
  - Need to compute a full distribution instead of a single point

Main families of general-purpose inference algorithms

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

(there are others)

Markov Chain Monte Carlo (MCMC)

- Generates a random sequence $\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \ldots$ (a Markov chain)

Markov Chain Monte Carlo (MCMC)

- Generates a random sequence $\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \ldots$ (a Markov chain)
- Various rules for drawing $\boldsymbol{\theta}_{n+1} | \boldsymbol{\theta}_n$ depending on the algorithm

Markov Chain Monte Carlo (MCMC)

- Generates a random sequence $\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \ldots$ (a Markov chain)
- Various rules for drawing $\boldsymbol{\theta}_{n+1} | \boldsymbol{\theta}_n$ depending on the algorithm
  - These will generally depend on $p(\boldsymbol{\theta}_n, \text{data})$, $p(\boldsymbol{\theta}_{n+1}, \text{data})$

Markov Chain Monte Carlo (MCMC)

- Generates a random sequence $\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \ldots$ (a Markov chain)
- Various rules for drawing $\boldsymbol{\theta}_{n+1} | \boldsymbol{\theta}_n$ depending on the algorithm
  - These will generally depend on $p(\boldsymbol{\theta}_n, \text{data})$, $p(\boldsymbol{\theta}_{n+1}, \text{data})$
- **Output:** A set of samples $\boldsymbol{\theta}_0, \ldots, \boldsymbol{\theta}_N$

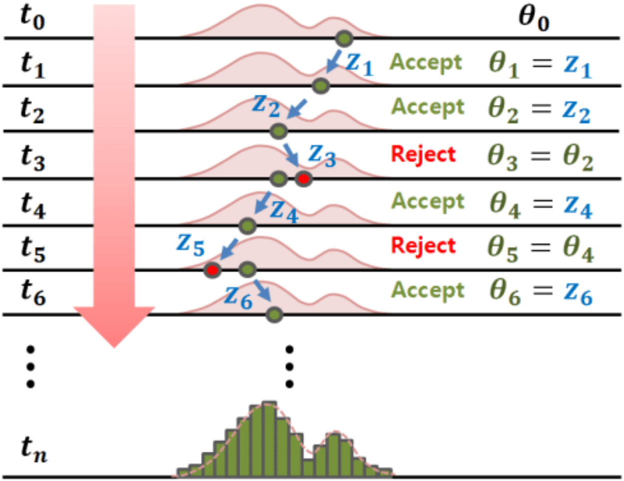
Markov Chain Monte Carlo (MCMC)

- Generates a random sequence $\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \ldots$ (a Markov chain)
- Various rules for drawing $\boldsymbol{\theta}_{n+1}|\boldsymbol{\theta}_n$ depending on the algorithm
  - These will generally depend on $p(\boldsymbol{\theta}_n, \text{data})$, $p(\boldsymbol{\theta}_{n+1}, \text{data})$
- **Output:** A set of samples $\boldsymbol{\theta}_0, \ldots, \boldsymbol{\theta}_N$
- **If all goes well**, $\boldsymbol{\theta}_0, \ldots, \boldsymbol{\theta}_N \sim p(\boldsymbol{\theta}|\text{data})$

Markov Chain Monte Carlo (MCMC)

- Generates a random sequence $\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \ldots$ (a Markov chain)
- Various rules for drawing $\boldsymbol{\theta}_{n+1} | \boldsymbol{\theta}_n$ depending on the algorithm
  - These will generally depend on $p(\boldsymbol{\theta}_n, \text{data})$, $p(\boldsymbol{\theta}_{n+1}, \text{data})$
- **Output:** A set of samples $\boldsymbol{\theta}_0, \ldots, \boldsymbol{\theta}_N$
- **If all goes well**, $\boldsymbol{\theta}_0, \ldots, \boldsymbol{\theta}_N \sim p(\boldsymbol{\theta} | \text{data})$
  - 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 $p(\boldsymbol{\theta}|\text{data})$ with $q_{\phi}(\boldsymbol{\theta})$

Variational inference

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

Outputs:

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

Variational inference

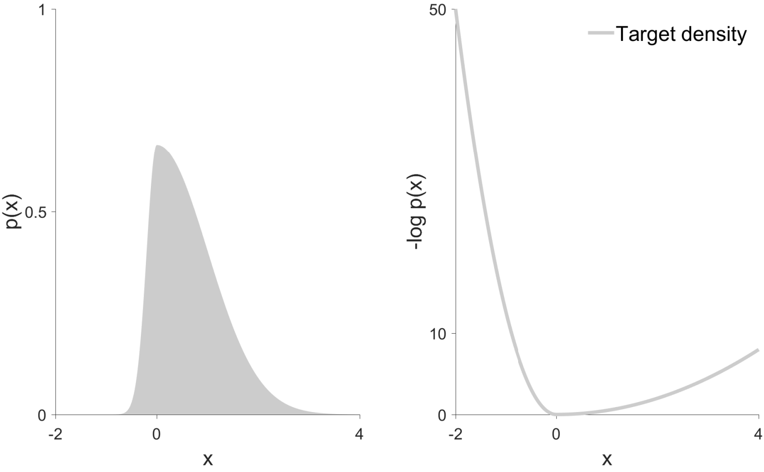
- Approximate $p(\boldsymbol{\theta}|\text{data})$ with $q_{\phi}(\boldsymbol{\theta})$
- Minimize Kullback-Leibler divergence between q and p

Outputs:

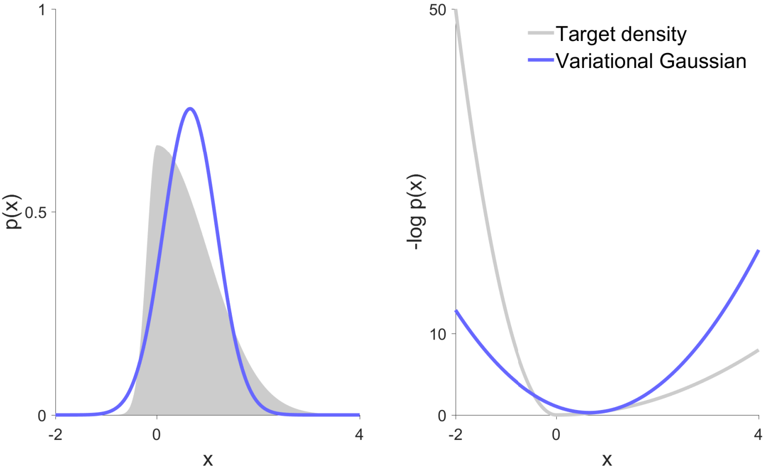
- An approximate posterior $q_{\phi}(\boldsymbol{\theta})$
- A lower bound to the log marginal likelihood, $\text{ELBO}(\phi)$

VI casts Bayesian inference into optimization + integration

Variational inference: example

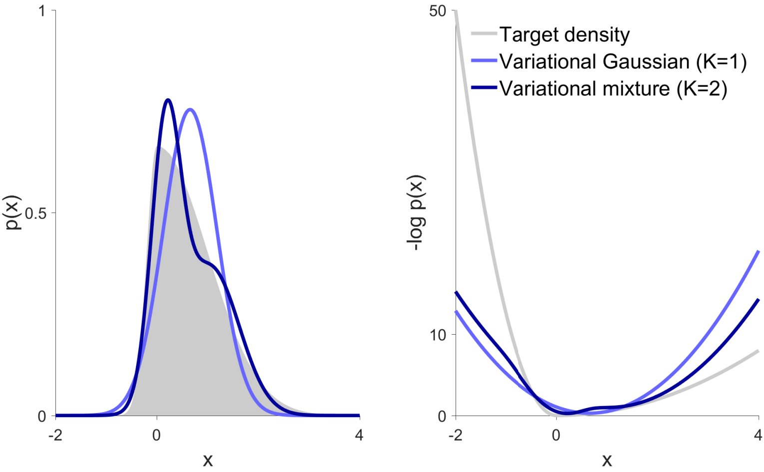


Variational inference: example



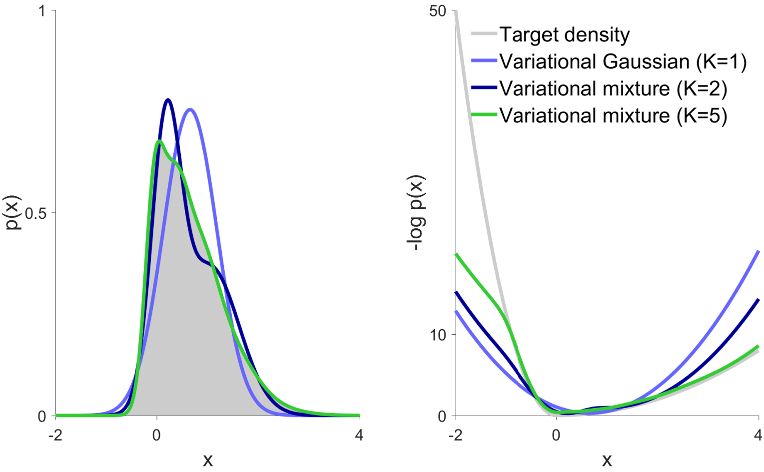
$$q_\phi(x) = \mathcal{N}\left(x, \mu, \sigma^2\right) \qquad \phi = \left(\mu, \sigma^2\right)$$

Variational inference: example



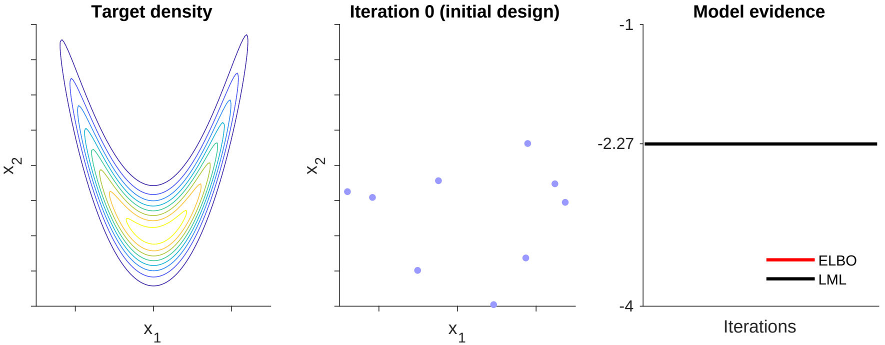
$$q_\phi(x) = \sum_{k=1}^{K} w_k \mathcal{N}\left(x, \mu_k, \sigma_k^2\right) \qquad \phi = \left(w_k, \mu_k, \sigma_k^2\right)_{k=1}^{K}$$

Variational inference: example



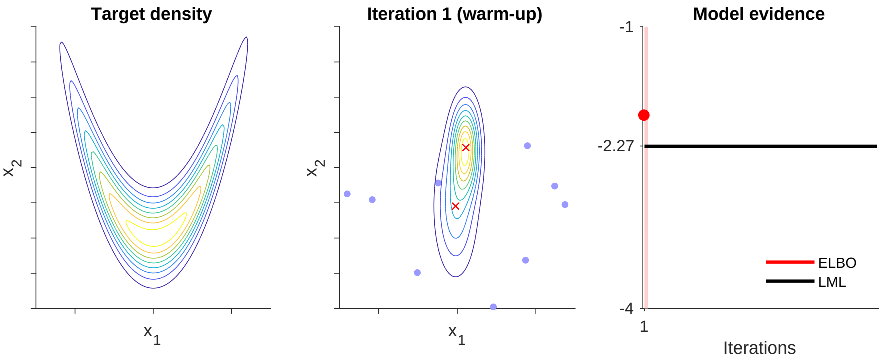
$$q_\phi(x) = \sum_{k=1}^{K} w_k \mathcal{N}\left(x, \mu_k, \sigma_k^2\right) \qquad \phi = \left(w_k, \mu_k, \sigma_k^2\right)_{k=1}^{K}$$

Variational Bayesian Monte Carlo (VBMC)



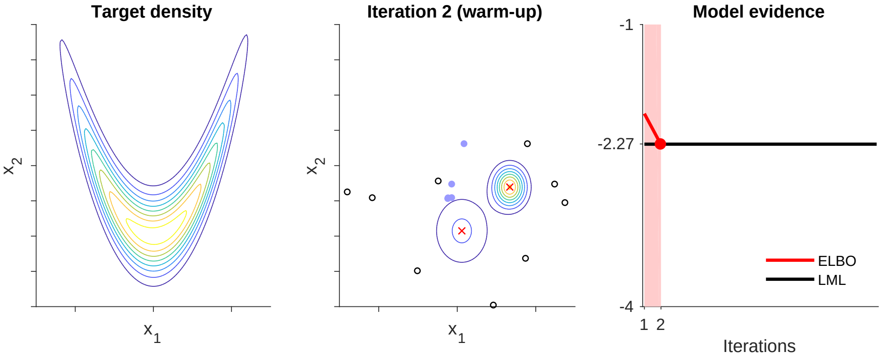
Acerbi, *NeurIPS* (2018; 2020)

Variational Bayesian Monte Carlo (VBMC)



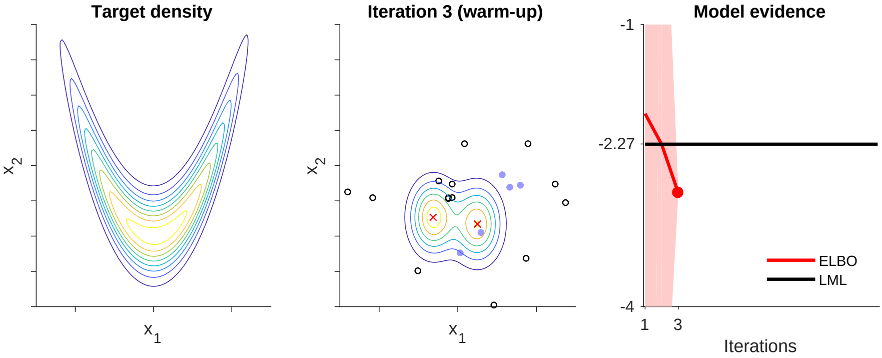
Acerbi, *NeurIPS* (2018; 2020)

Variational Bayesian Monte Carlo (VBMC)



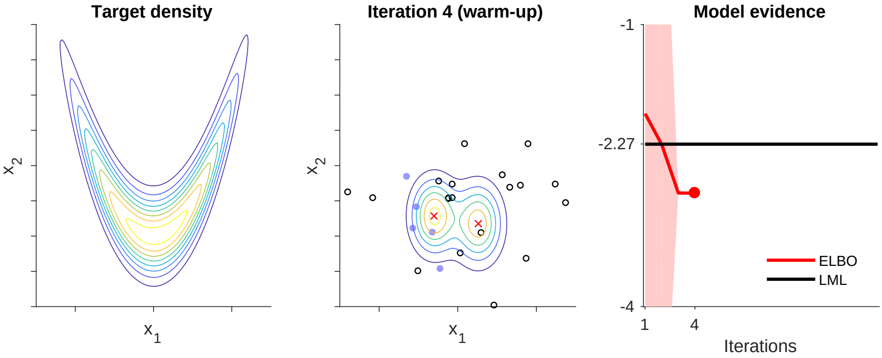
Acerbi, *NeurIPS* (2018; 2020)

Variational Bayesian Monte Carlo (VBMC)



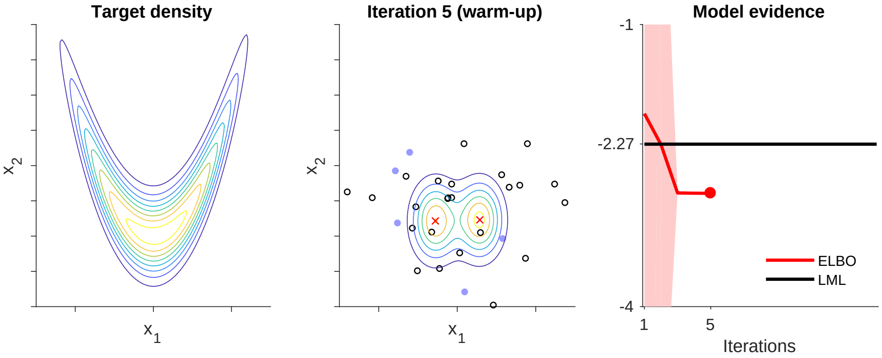
Acerbi, *NeurIPS* (2018; 2020)

Variational Bayesian Monte Carlo (VBMC)



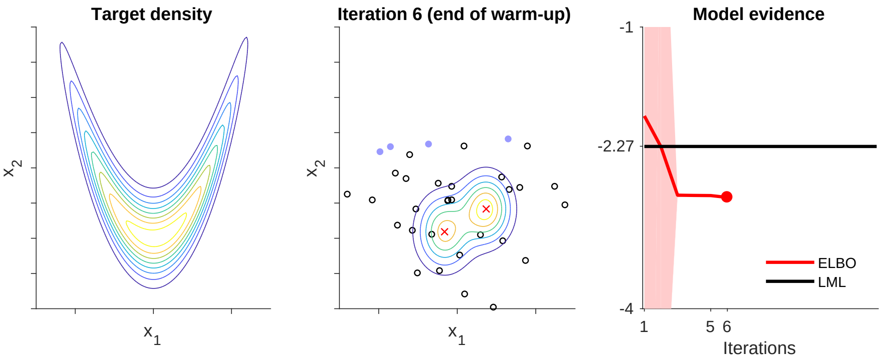
Acerbi, *NeurIPS* (2018; 2020)

Variational Bayesian Monte Carlo (VBMC)



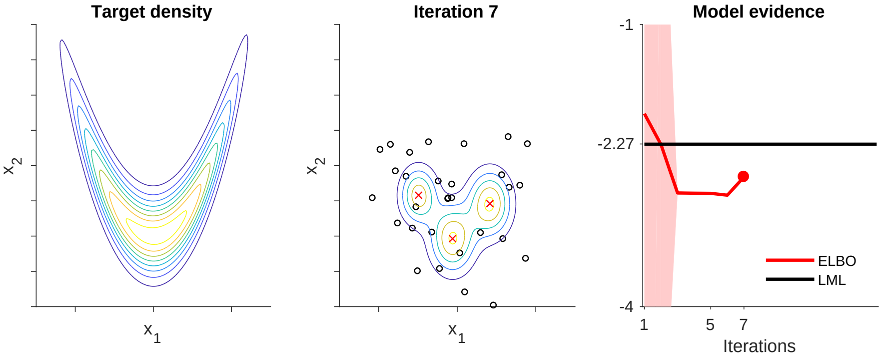
Acerbi, *NeurIPS* (2018; 2020)

Variational Bayesian Monte Carlo (VBMC)



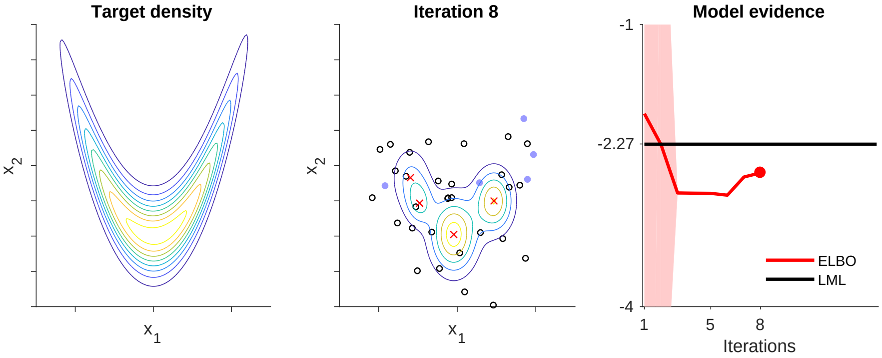
Acerbi, *NeurIPS* (2018; 2020)

Variational Bayesian Monte Carlo (VBMC)



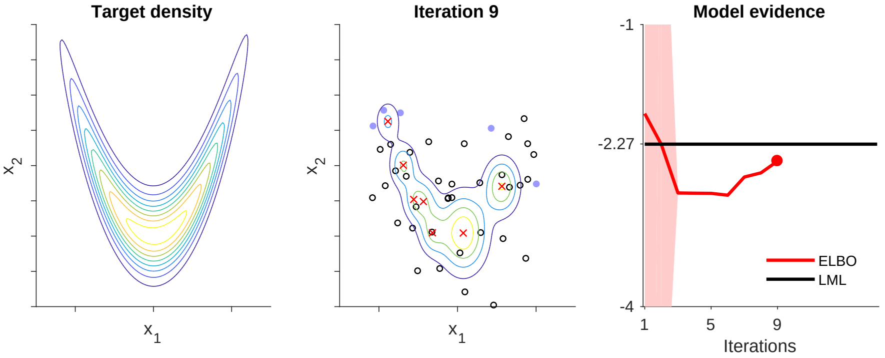
Acerbi, *NeurIPS* (2018; 2020)

Variational Bayesian Monte Carlo (VBMC)



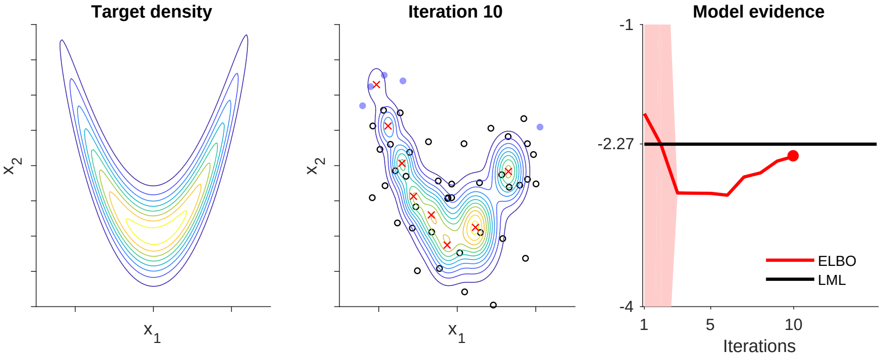
Acerbi, *NeurIPS* (2018; 2020)

Variational Bayesian Monte Carlo (VBMC)



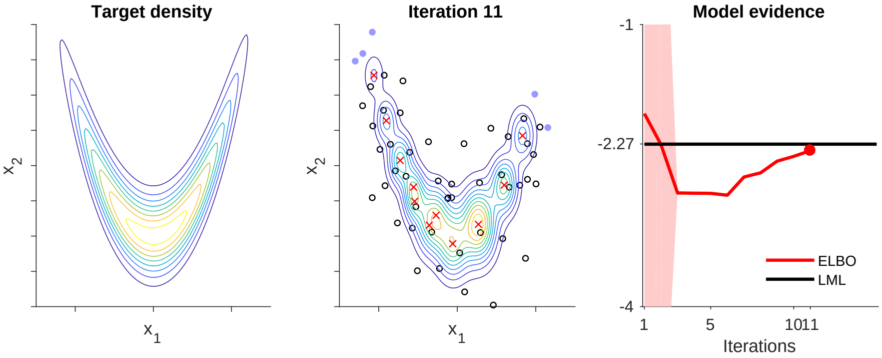
Acerbi, *NeurIPS* (2018; 2020)

Variational Bayesian Monte Carlo (VBMC)



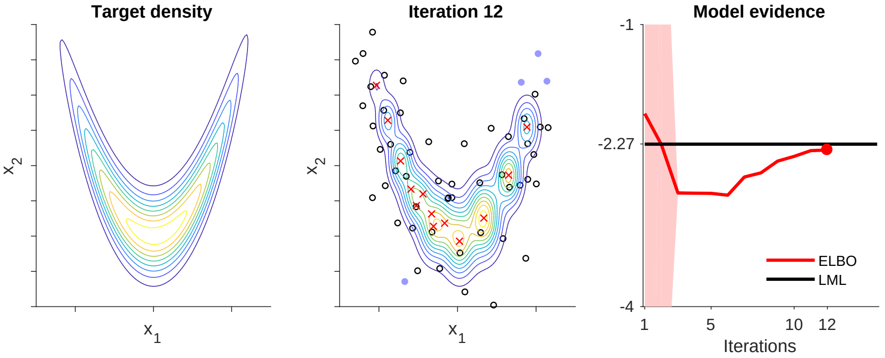
Acerbi, *NeurIPS* (2018; 2020)

Variational Bayesian Monte Carlo (VBMC)



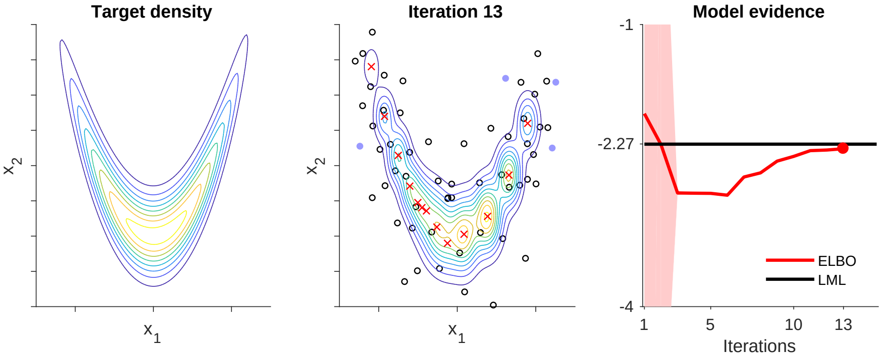
Acerbi, *NeurIPS* (2018; 2020)

Variational Bayesian Monte Carlo (VBMC)



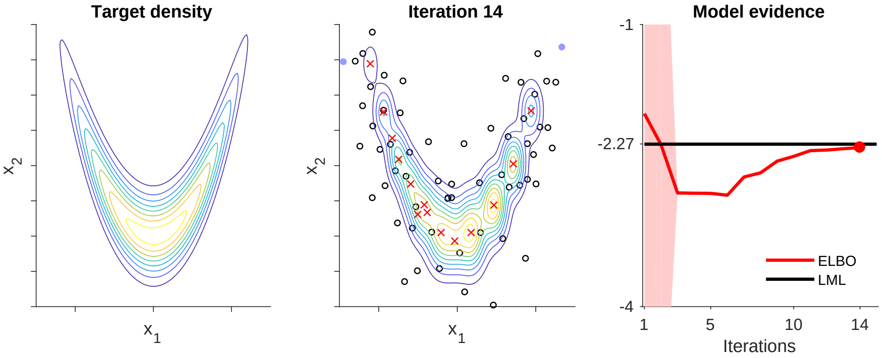
Acerbi, *NeurIPS* (2018; 2020)

Variational Bayesian Monte Carlo (VBMC)



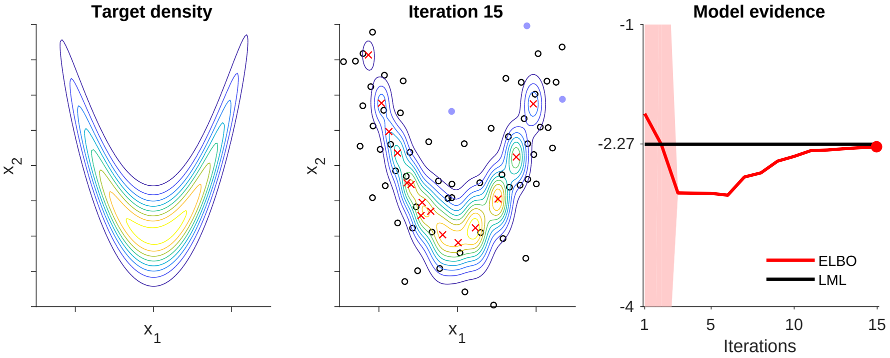
Acerbi, *NeurIPS* (2018; 2020)

Variational Bayesian Monte Carlo (VBMC)



Acerbi, *NeurIPS* (2018; 2020)

Variational Bayesian Monte Carlo (VBMC)



Acerbi, *NeurIPS* (2018; 2020)

Hacking time III

Let's set up and run a Bayesian inference algorithm

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?)