Trends in Computational Neuroscience: Introduction to Statistical Model Fitting

Luigi Acerbi

Department of Basic Neuroscience, University of Geneva
Center for Neural Science, New York University
International Brain Lab

INTERNATIONAL
BRAIN
LABORATORY

April 2020

What is a model?

The best material model of a cat is another, or preferably the same, cat.

Wiener, *Philosophy of Science* (1945) (with Rosenblueth)

What is a mathematical model?

- Quantitative stand-in for a theory
- A *family of probability distributions* over possible datasets:

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

  - data is a dataset with n data points (e.g., trials)
  - $\boldsymbol{\theta}$ is a parameter vector
- **Why?** Description, prediction, and explanation
- Defining $p(\text{data}|\boldsymbol{\theta})$ is the core of model building
  - Wait, what?
- **How?** Think about the data generation process!

We need some data

Intermission: International Brain Laboratory

Neuron
NeuroView

CellPress

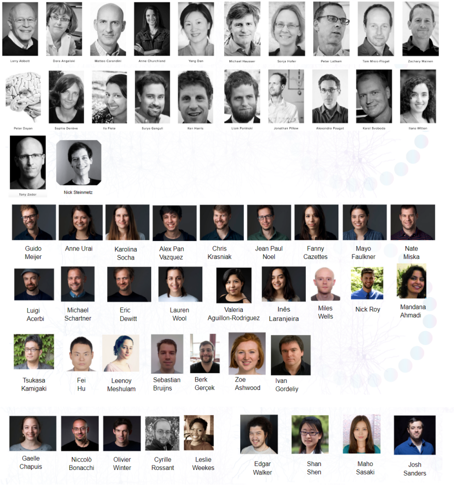
An International Laboratory for Systems and Computational Neuroscience

The International Brain Laboratory*
*Correspondence: churchland@cshl.edu
https://doi.org/10.1016/j.neuron.2017.12.013

The neural basis of decision-making has been elusive and involves the coordinated activity of multiple brain structures. This NeuroView, by the International Brain Laboratory (IBL), discusses their efforts to develop a standardized mouse decision-making behavior, to make coordinated measurements of neural activity across the mouse brain, and to use theory and analyses to uncover the neural computations that support decision-making.

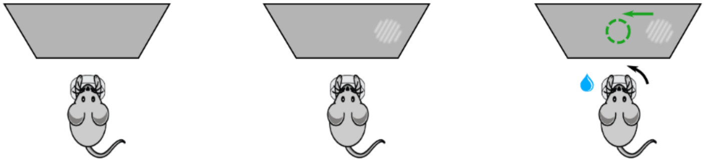
(IBL, *Neuron*, 2017)

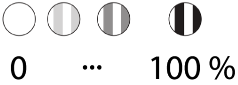
Intermission: International Brain Laboratory



https://www.internationalbrainlab.com

IBL Task





Hacking time I

Let's have a look at the data

Type of models

- Descriptive
- Mechanistic
- Process
- Normative
- ...

ORIGINAL ARTICLE

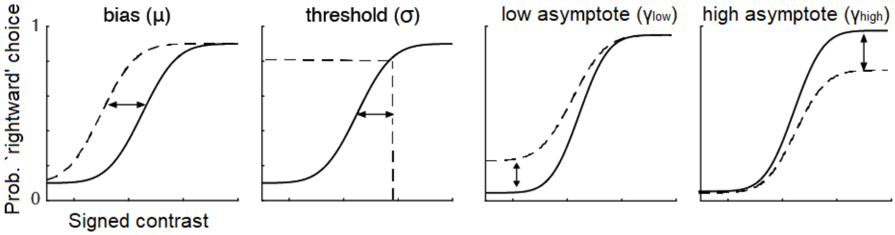
Commentary

Appreciating the variety of goals in computational neuroscience

Konrad P. Kording PhD[1] | Gunnar Blohm PhD[2] | Paul Schrater PhD[3] | Kendrick Kay PhD[4]

https://arxiv.org/abs/2002.03211

The psychometric function



- Data: (signed contrast, choice) for each trial
- Parameters $\boldsymbol{\theta}$: $(\mu, \sigma, \gamma_{\text{low}}, \gamma_{\text{high}})$

$$p(\text{rightward choice}|s, \boldsymbol{\theta}) = \gamma_{\text{low}} + (1 - \gamma_{\text{low}} - \gamma_{\text{high}}) \cdot F(s; \mu, \sigma)$$

The psychometric function (alt version)

- Default decision process $F(s; \mu, \sigma)$
- Lapses with probability $\lambda \in [0, 1]$ (*lapse rate*)
- If lapse, respond 'rightward' with probability $\gamma \in [0, 1]$ (*lapse bias*)
- Parameters $\boldsymbol{\theta}$: $(\mu,\ \sigma,\ \lambda,\ \gamma)$

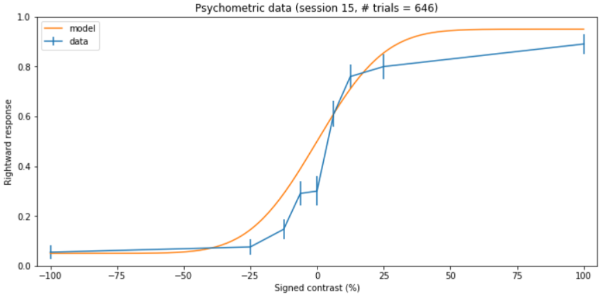
$$p(\text{rightward choice}|s, \boldsymbol{\theta}) = \lambda\gamma + (1 - \lambda) \cdot F(s; \mu, \sigma)$$

Hacking time II

Let's code a psychometric function

Metric for model fitting

We need a quantity to measure *goodness of fit*



- ~~Mean squared error?~~
- The likelihood $p(\text{data}|\boldsymbol{\theta})$

The (log) likelihood

- $p(\text{data}|\boldsymbol{\theta})$ is a *probability density* as you vary data for a fixed $\boldsymbol{\theta}$
- $p(\text{data}|\boldsymbol{\theta})$ is the *likelihood*, a function of $\boldsymbol{\theta}$ for fixed data

- For numerical reasons we work with $\log p(\text{data}|\boldsymbol{\theta})$
- Using the rules of probability and logarithms:

$$\begin{aligned}\log p\left(\text{data}|\boldsymbol{\theta}\right) &= \log p(\boldsymbol{r}^{(1)}, \ldots, \boldsymbol{r}^{(n)}|\boldsymbol{s}^{(1)}, \ldots, \boldsymbol{s}^{(n)}, \boldsymbol{\theta}) \\ &= \log \prod_{i=1}^{n} p_i \left(\boldsymbol{r}^{(i)}|\boldsymbol{r}^{(1)}, \ldots, \boldsymbol{r}^{(i-1)}, \boldsymbol{s}^{(1)}, \ldots, \boldsymbol{s}^{(n)}, \boldsymbol{\theta}\right) \\ &= \sum_{i=1}^{n} \log p_i \left(\boldsymbol{r}^{(i)}|\boldsymbol{r}^{(1)}, \ldots, \boldsymbol{r}^{(i-1)}, \boldsymbol{s}^{(1)}, \ldots, \boldsymbol{s}^{(n)}, \boldsymbol{\theta}\right)\end{aligned}$$

- Simplest case: $\log p\left(\text{data}|\boldsymbol{\theta}\right) = \sum_{i=1}^{n} \log p_i \left(\boldsymbol{r}^{(i)}|\boldsymbol{s}^{(i)}, \boldsymbol{\theta}\right)$
- Model building: Write function with
  - Input: $\boldsymbol{\theta}$ and data
  - Output: $\log p(\text{data}|\boldsymbol{\theta})$

Hacking time III

Let's code up a log-likelihood function

Model fitting

Model fitting $\sim$ *statistical estimation* problem

1. Maximum likelihood estimation (MLE)

- Find maximum of $p(\text{data}|\boldsymbol{\theta})$

$$\hat{\boldsymbol{\theta}}_{\text{ML}} = \arg\max_{\boldsymbol{\theta}} p(\text{data}|\boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} \log p(\text{data}|\boldsymbol{\theta})$$

2. Bayesian posterior

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

- For $n \to \infty$ converges to MLE (if $p(\hat{\boldsymbol{\theta}}_{\text{ML}}) \neq 0$)
- *Full posterior*: informative about parameter uncertainty and trade-offs
- *Maximum-a-posteriori (MAP)*: $\hat{\boldsymbol{\theta}}_{\text{MAP}} = \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta}|\text{data})$

How to do model fitting?

Maximum likelihood estimation (MLE), Maximum-a-posteriori (MAP)

- Model fitting $\sim$ *optimization problem*

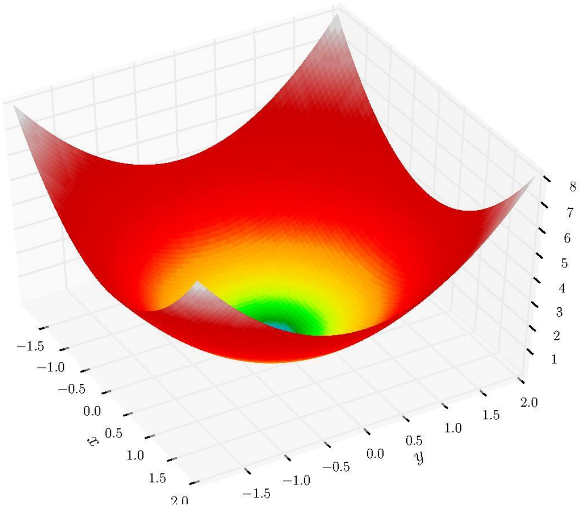
Bayesian posterior

- How do we represent/approximate an arbitrary posterior distribution?
  1. Use a known (easier) distribution (*variational inference*)
  2. Use a bunch of discrete samples (*Markov-Chain Monte Carlo*)

Model fitting via optimization

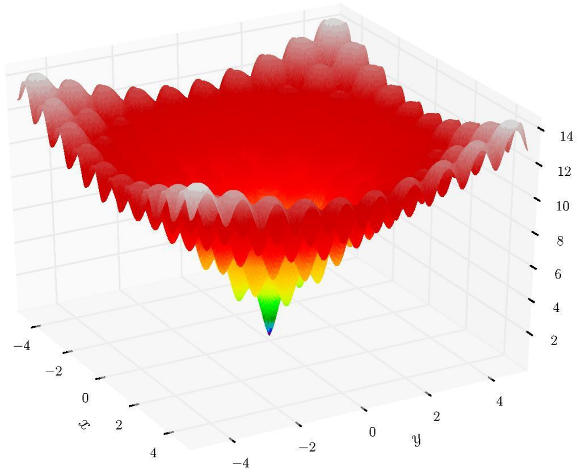
- Find single $\boldsymbol{\theta}$ that best describes the data
- (For this section we switch notation from $\boldsymbol{\theta}$ to $\boldsymbol{x}$)
- Given $\tilde{f}(\boldsymbol{x}) \equiv \begin{cases} \log p(\text{data}|\boldsymbol{x}) & \text{maximum likelihood} \\ \log p(\text{data}|\boldsymbol{x}) + \log p(\boldsymbol{x}) & \text{maximum-a-posteriori} \end{cases}$
- By convention, we *minimize* $f(x) \equiv -\tilde{f}(\boldsymbol{x})$
- $\Longrightarrow$ Find $\boldsymbol{x}_{opt} \approx \arg\min_{\boldsymbol{x}} f(\boldsymbol{x})$ as fast as possible
- General case: $f(\boldsymbol{x})$ is a *black box*
  - Sometimes we can compute the gradient

How hard can it be?



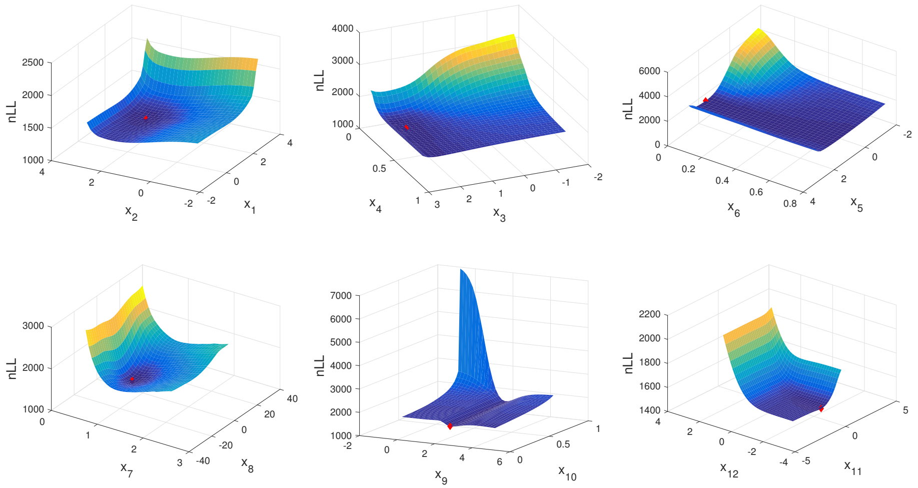
Source: Wikimedia Commons

How hard can it be?

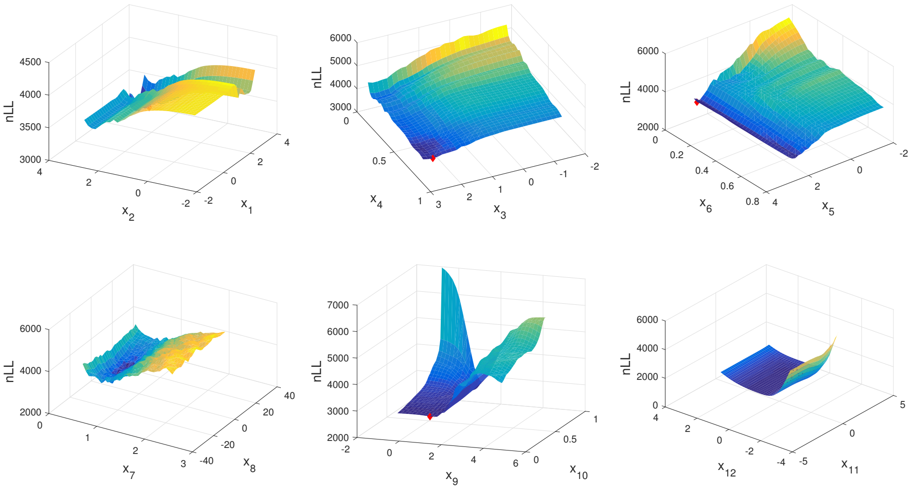


Source: Wikimedia Commons

How hard can it be?



How hard can it be?



How hard can it be?

neval	x_1	x_2	$f(\boldsymbol{x})$
1	-0.500	2.500	508.500
2	-0.525	2.500	497.110
3	-0.500	2.625	566.313
4	-0.525	2.375	443.063
5	-0.537	2.250	386.953
6	-0.563	2.250	376.320
7	-0.594	2.125	316.702
8	-0.606	1.875	229.824
9	-0.647	1.563	133.598
10	-0.703	1.438	91.847
11	-0.786	1.031	20.292
12	-0.839	0.469	8.918
13	-0.962	-0.359	168.785
14	-0.978	-0.063	107.796
15	-0.895	0.344	24.553
16	-0.730	1.156	41.905
17	-0.854	0.547	6.760
18	-0.907	-0.016	73.917
19	-0.816	0.770	4.366
20	-0.831	0.848	5.818
21	-0.793	1.070	22.655
22	-0.839	0.678	3.448
23	-0.824	0.600	3.955
24	-0.846	0.508	7.766
25	-0.824	0.704	3.391
26	-0.839	0.782	4.004
27	-0.828	0.645	3.497
28	-0.835	0.737	3.523
29	?	?	?

Optimization can be hard

1. Optimizer does not see the landscape!
2. Multiple local minima or saddle points (‘non-convex’)
3. Expensive function evaluation
4. Noisy function evaluation
5. Rough landscape (numerical approximations, etc.)

Optimization algorithms

Gradient-based methods

- Stochastic gradient descent (e.g., ADAM)
- Quasi-Newton methods (e.g., BFGS aka `fminunc`/`fmincon`)

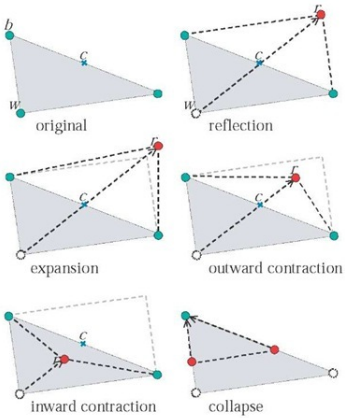
Gradient-free methods

- Nelder-Mead (`fminsearch`)
- Pattern/direct search (`patternsearch`)
- Simulated annealing
- Genetic algorithms
- CMA-ES
- Bayesian optimization
- Bayesian Adaptive Direct Search (BADS; Acerbi & Ma, *NeurIPS* 2017)

Demos: `https://github.com/lacerbi/optimviz`

Nelder-Mead (`fminsearch`)

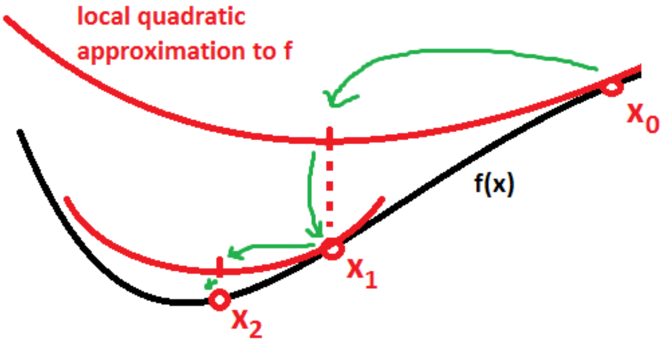
J. A. Nelder & R. Mead, A simplex method for function minimization (1965)



Source: Encyclopedia of Artificial Intelligence (2009)

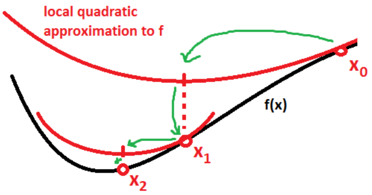
Bounded optimization: `fminsearchbnd` (John d'Errico)

Newton method



Source: StackExchange

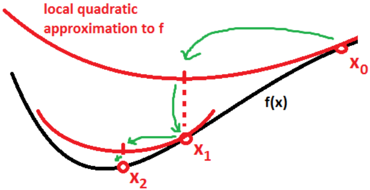
Newton method



Source: StackExchange

Needs the inverse of the curvature (inverse Hessian)
Very expensive in high dimension

Quasi-Newton methods (`fminunc`,`fmincon`)

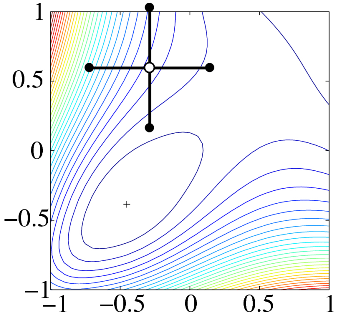


Source: StackExchange

Approximate Hessian (DFP) or inverse Hessian (BFGS) via gradient
Very fast and efficient on smooth problems

Direct search (`patternsearch`)

R. Hooke and T.A. Jeeves, "Direct search" solution of numerical and statistical problems (1961)

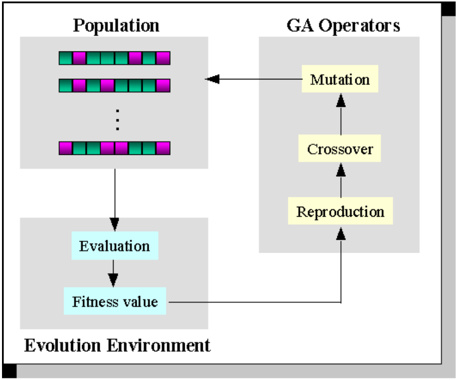


Source: Wikimedia Commons

Genetic Algorithms (ga)

J.H. Holland, Adaptation in Natural and Artificial Systems (1975)

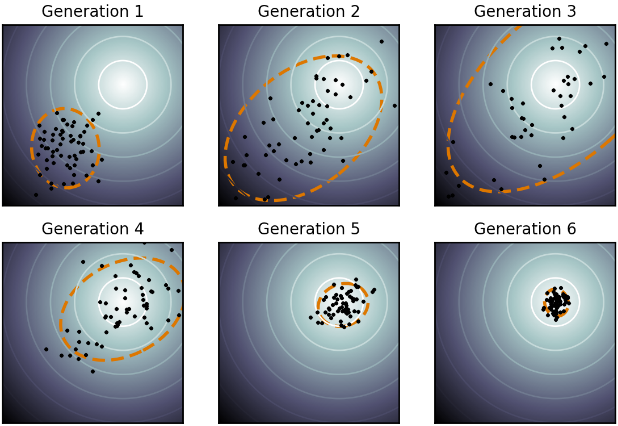
- Evolutionary algorithm
- Population based



Source: An Educational GA Learning Tool (IEEE)

Cov. Matrix Adaptation - Evolution Strategies (`cmaes`)

[*] N. Hansen, S. D. Müller, P. Koumoutsakos, Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES), (2003)



Source: Wikipedia

Hacking time IV

Let's optimize the log-likelihood for the psychometric model

Local vs. global optimization

Local ↔ Global

BFGS (`fmincon`)	Nelder-Mead (`fminsearch`)	BADS	Bayesian Optimization
	`patternsearch`	CMA-ES	MCS
	SGD	Genetic algorithms	
		Simulated annealing	

Optimization cheat sheet, page 1

Rule zero

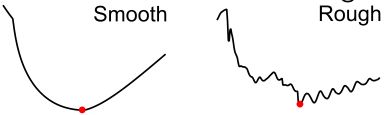
Understand your problem $\Longrightarrow$ often a *gray* box

Input variables:

- Dimensionality: low ($D \lesssim 10$) or high ($D \gg 20$)
- Bounds: Think of *hard* and *plausible* bounds
- Parameterization: Not all parameterizations are created equal

Target function:

- Convexity: convex or non-convex
- Smoothness: smooth or rough



- Deterministic or stochastic
  - If stochastic $\Longrightarrow$ minimize $\mathbb{E}\left[f(\boldsymbol{x})\right]$
- Computational cost:
  cheap ($\ll 0.01$ s), moderate (0.01-1 s), or expensive ($\gg 1$ s)

Optimization cheat sheet, page 2

Fundamental theorem

'No Free Lunch' theorem $\Longrightarrow$ no single best optimizer for all problems
(But not all methods are created equal!)

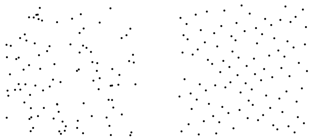
- Is your problem smooth?
  - If you have the gradient $\Longrightarrow$ BFGS
  - If low-D and cheap $\Longrightarrow$ BFGS with finite differences
  - If low-D and (moderately) costly $\Longrightarrow$ BADS
- Is your problem rough or noisy?
  - First, try and make it smooth and deterministic!
  - If gradient is available and high-D $\Longrightarrow$ SGD (e.g., ADAM)
  - If high-D and cheap $\Longrightarrow$ CMA-ES
  - If low-D and (moderately) costly $\Longrightarrow$ BADS
- Is your problem high-D, costly, and you do not have the gradient?
  - Give up and pray

Optimization cheat sheet, page 3

The golden rule

No optimizer can guarantee to find the global optimum
$\Longrightarrow$ Always perform multiple distinct optimization runs ('restarts')

- How to choose starting points?
  - Draw from prior distribution
  - Draw from a 'plausible' box
  - Sieve method
  - Use space-filling designs (quasi-random sequences)



- How many restarts?
  - As many as you need
  - Informally, check that 'most' points converge to the same solution
  - *Bootstrap* approach (Acerbi, Dokka et al., *PLoS Comp Biol* 2018)

1. Introduction
   - Of models and likelihoods
   - The psychometric function
2. Model fitting
   - A statistical estimation problem
   - Model fitting via optimization
   - Optimization algorithms
   - Optimization cheat sheets
3. **Model selection**
   - Information criteria (AIC/BIC)
   - Cross-validation (CV)
   - Marginal likelihood and Laplace approximation
4. Advanced (Bayesian) topics
   - Bayesian Optimization
   - Bayesian model fitting

The problem

- Several models $\mathcal{M}_1, \ldots, \mathcal{M}_M$
- For each $\mathcal{M}_m$ we know $\log p(\text{data}|\hat{\theta}_{\text{ML}}, \mathcal{M}_m)$
- Find the *best* model

Typical form of model comparison metric

$$MCM(\text{data}, \mathcal{M}_m) \propto \overbrace{\log p(\text{data}|\hat{\theta}_{\text{ML}}, \mathcal{M}_m)}^{\text{Goodness of fit}} - \overbrace{f\left(\text{data}, \mathcal{M}_m\right)}^{\text{Model complexity}}$$

Notation:

- k number of parameters
- n number of trials

Akaike information criterion (AIC)

Akaike information criterion

$$\mathrm{AIC} = \log p(\mathrm{data}|\hat{\theta}_{\mathrm{ML}}, \mathcal{M}_m) - k$$

Akaike information criterion (AIC)

Akaike information criterion

$$\mathrm{AIC} = -2 \log p(\mathrm{data}|\hat{\theta}_{\mathrm{ML}}, \mathcal{M}_m) + 2k$$

Akaike information criterion (AIC)

Akaike information criterion

$$\text{AIC} = -2 \log p(\text{data}|\hat{\theta}_{\text{ML}}, \mathcal{M}_m) + 2k$$

- **Goal:** Find best predictive model
  - Does not assume $\mathcal{M}_{\text{true}}$ is in the model set
  - Find closest statistical approximation (lowest KL-divergence from $\mathcal{M}_{\text{true}}$)

Akaike information criterion (AIC), part two

Why penalty is k?

- Same thing as $\mathcal{M}_m$ that *minimizes* $KL(p_{\text{true}}||p_m)$
- $\left\langle \log p(y|\hat{\boldsymbol{\theta}}_{\text{ML}}, \mathcal{M}_m) \right\rangle_{y \sim p_{\text{true}}} \approx \frac{1}{n} \sum_{i=1}^{n} \log p(y_i|\hat{\boldsymbol{\theta}}_{\text{ML}}, \mathcal{M}_m)$
- $\frac{1}{n} \sum_{i=1}^{n} \log p(y_i|\hat{\boldsymbol{\theta}}_{\text{ML}}, \mathcal{M}_m)$ is a *biased* estimate
- Bias correction per trial $\approx \frac{1}{n} k$
- Assumptions:
  - CLT (large n), log likelihood $\sim$ quadratic near MLE
  - p close to p_{true}
  - model identifiable (bijective mapping $\boldsymbol{\theta} \longleftrightarrow p(y|\boldsymbol{\theta})$)

Akaike information criterion (AIC), part two

Why penalty is k?

(Do you really want to know?)

- Same thing as $\mathcal{M}_m$ that *minimizes* $KL(p_{\text{true}}||p_m)$
- $\left\langle \log p(y|\hat{\boldsymbol{\theta}}_{\text{ML}}, \mathcal{M}_m) \right\rangle_{y \sim p_{\text{true}}} \approx \frac{1}{n} \sum_{i=1}^{n} \log p(y_i|\hat{\boldsymbol{\theta}}_{\text{ML}}, \mathcal{M}_m)$
- $\frac{1}{n} \sum_{i=1}^{n} \log p(y_i|\hat{\boldsymbol{\theta}}_{\text{ML}}, \mathcal{M}_m)$ is a *biased* estimate
- Bias correction per trial $\approx \frac{1}{n} k$
- Assumptions:
  - CLT (large n), log likelihood $\sim$ quadratic near MLE
  - p close to p_{true}
  - model identifiable (bijective mapping $\boldsymbol{\theta} \longleftrightarrow p(y|\boldsymbol{\theta})$)

Akaike information criterion (AIC), part two

Why penalty is k?

Best predictive model

$$\mathcal{M}_m \text{ that maximizes } \left\langle \log p(y|\hat{\boldsymbol{\theta}}_{\mathrm{ML}}, \mathcal{M}_m) \right\rangle_{y \sim p_{\mathrm{true}}}$$

- Same thing as $\mathcal{M}_m$ that *minimizes* $KL(p_{\mathrm{true}}||p_m)$
- $\left\langle \log p(y|\hat{\boldsymbol{\theta}}_{\mathrm{ML}}, \mathcal{M}_m) \right\rangle_{y \sim p_{\mathrm{true}}} \approx \frac{1}{n} \sum_{i=1}^{n} \log p(y_i|\hat{\boldsymbol{\theta}}_{\mathrm{ML}}, \mathcal{M}_m)$
- $\frac{1}{n} \sum_{i=1}^{n} \log p(y_i|\hat{\boldsymbol{\theta}}_{\mathrm{ML}}, \mathcal{M}_m)$ is a *biased* estimate
- Bias correction per trial $\approx \frac{1}{n} k$
- Assumptions:
  - CLT (large n), log likelihood $\sim$ quadratic near MLE
  - p close to p_{true}
  - model identifiable (bijective mapping $\boldsymbol{\theta} \longleftrightarrow p(y|\boldsymbol{\theta})$)

Schwarz (Bayesian) information criterion (BIC)

Bayesian information criterion

$$\mathrm{BIC} = \log p(\mathrm{data}|\hat{\theta}_{\mathrm{ML}}, \mathcal{M}_m) - \frac{1}{2} k \log n$$

- **Goal:** Find true model
  - Assume $\mathcal{M}_{\mathrm{true}}$ is in the model set
  - Based on loooose approximation of $P(\mathcal{M}|\mathrm{data})$
- Penalizes complexity much more than AIC(c)
- *Consistent:* for $n \to \infty$ selects $\mathcal{M}_{\mathrm{true}}$ if $\mathcal{M}_{\mathrm{true}}$ in model set

Schwarz (Bayesian) information criterion (BIC)

Bayesian information criterion

$$\mathrm{BIC} = -2 \log p(\mathrm{data}|\hat{\theta}_{\mathrm{ML}}, \mathcal{M}_m) + k \log n$$

- **Goal:** Find true model
  - Assume $\mathcal{M}_{\mathrm{true}}$ is in the model set
  - Based on loooose approximation of $P(\mathcal{M}|\mathrm{data})$
- Penalizes complexity much more than AIC(c)
- *Consistent:* for $n \to \infty$ selects $\mathcal{M}_{\mathrm{true}}$ if $\mathcal{M}_{\mathrm{true}}$ in model set

Cross-validation

- **Goal:** Find best predictive model
  - Split data in training and validation
  - $\frac{1}{n} \langle \log p(\text{data}|\boldsymbol{\theta}_{\text{ML}}, \mathcal{M}_m) \rangle_{p_{\text{true}}} \approx \left\langle \frac{1}{n_V} \log p(\text{validation data}|\hat{\boldsymbol{\theta}}_{\text{train}}, \mathcal{M}_m) \right\rangle_{\text{train, validation}}$

Cross-validated log likelihood

$$\text{CV} = \frac{1}{K} \sum_{i=1}^{K} \frac{1}{n_V} \log p(\text{validation data}^{(i)}|\hat{\boldsymbol{\theta}}_{\text{train}^{(i)}}, \mathcal{M}_m)$$

- Typical cases: K-fold cross-validation, leave-one-out (LOO) cross-validation
  - AIC tends to LOO
- Essentially no assumptions (but caveats)
- Computationally expensive

Marginal likelihood

Can we be more Bayesian?

- **Goal:** Find model with highest posterior probability
  - $P(\mathcal{M}|\text{data}) = \frac{P(\text{data}|\mathcal{M})P(\mathcal{M})}{P(\text{data})}$

Marginal likelihood

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

- Pros: Theoretically sound, consistent, Bayesian Occam's razor
- Cons: Hard to compute, depends on choice of prior
- Laplace approximation:
  $P(\text{data}|\mathcal{M}) \approx \log p(\text{data}|\hat{\boldsymbol{\theta}}_{\text{ML}}, \mathcal{M}_m) + \frac{k}{2}\log 2\pi - \frac{1}{2}\log|\det \boldsymbol{H}(\boldsymbol{\theta}_{\text{ML}})|$
  - Can be good or terrible, depending on posterior and on the basis

Marginal likelihood

Can we be more Bayesian?

(Not really, with only point estimates)

- **Goal:** Find model with highest posterior probability
  - $P(\mathcal{M}|\text{data}) = \frac{P(\text{data}|\mathcal{M})P(\mathcal{M})}{P(\text{data})}$

Marginal likelihood

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

- Pros: Theoretically sound, consistent, Bayesian Occam's razor
- Cons: Hard to compute, depends on choice of prior
- Laplace approximation:
  $P(\text{data}|\mathcal{M}) \approx \log p(\text{data}|\hat{\boldsymbol{\theta}}_{\text{ML}}, \mathcal{M}_m) + \frac{k}{2}\log 2\pi - \frac{1}{2}\log|\det \boldsymbol{H}(\boldsymbol{\theta}_{\text{ML}})|$
  - Can be good or terrible, depending on posterior and on the basis

Model selection: The take-home slide

- AIC(c) vs BIC
  - AIC(c) will often pick the most complex model
  - BIC has too large penalty for complexity
  - Correct model complexity penalty often between AIC(c) and BIC
  - AIC(c) and BIC have no knowledge of the model
  - **Rule of thumb:** Try both; if they disagree, use more complex method
- Marginal likelihood
  - If you can compute it (analytically or numerically), use it
  - Laplace approximation may be okay for large n, but be careful
- Cross-validation (CV)
  - Takes into account structure of the model/parameters
  - Most common: 10-fold CV, or leave-one-out CV
  - Computationally expensive but might be worth it
  - Leave-one-out CV can be computed easily in some cases

Hacking time V

Let's compare some models

Bayesian Optimization

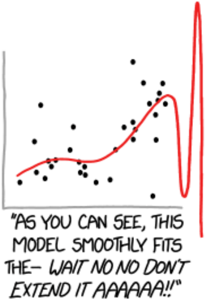
1. Start with a prior over functions (Gaussian process)
2. Find $\tilde{\boldsymbol{x}}$ that maximizes *acquisition function* (exploration/exploitation)
3. Evaluate $f(\tilde{\boldsymbol{x}})$
4. Compute posterior over functions (Gaussian process)
5. `goto 2`

J. Mockus, *Journal of Global Optimization* (1994)

Review paper: Frazier (2018) `https://arxiv.org/abs/1807.02811`

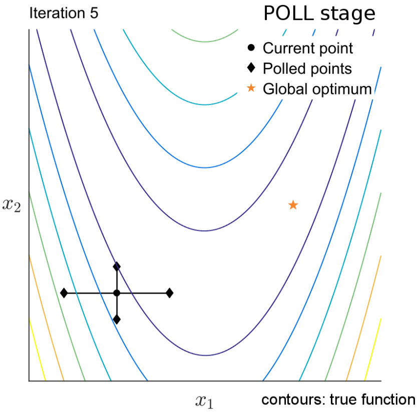
Why don't we use Bayesian optimization *all the time*?

- Matrix inversion is $O(n^3)$
- Model mismatch



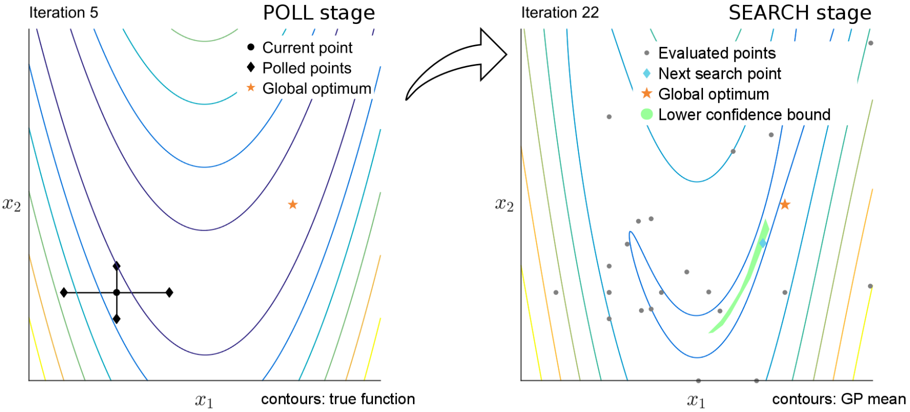
from `xkcd.com/2048`

Bayesian Adaptive Direct Search (bads)



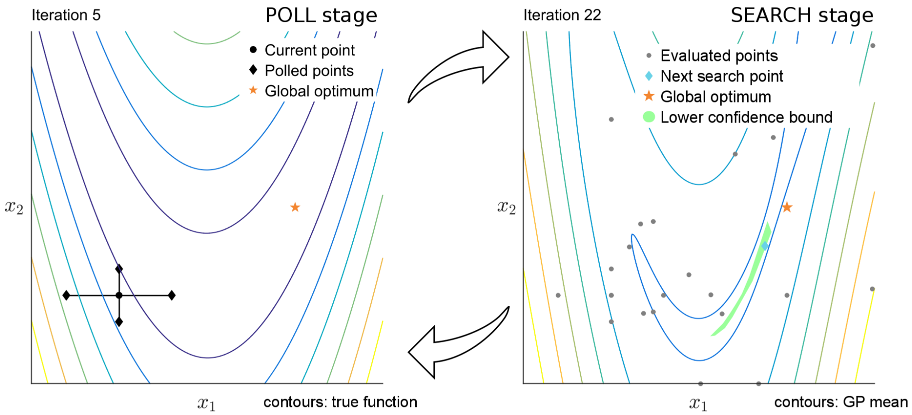
Acerbi & Ma, *NeurIPS* (2017)

Bayesian Adaptive Direct Search (bads)



Acerbi & Ma, *NeurIPS* (2017)

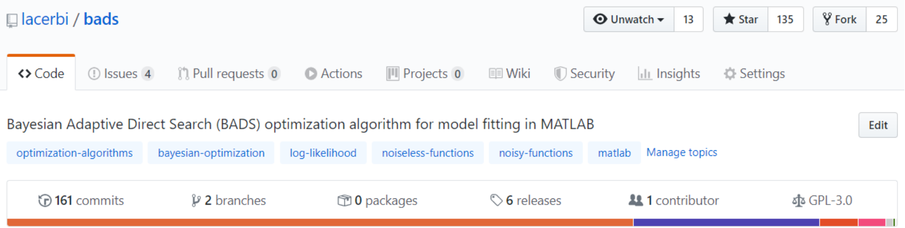
Bayesian Adaptive Direct Search (bads)



Acerbi & Ma, *NeurIPS* (2017)

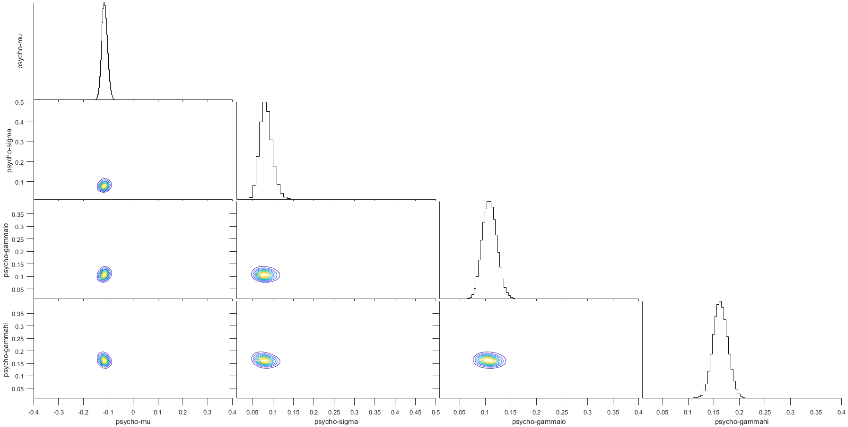
BADS summary

- Good for moderately costly ($\gtrsim 0.1$ s) or noisy functions
- Scales okay with n (uses only local neighborhood)
- Local approximation deals with nonstationarity
- Explicit support for noise
- Outperforms other algorithms (Acerbi & Ma, 2017)



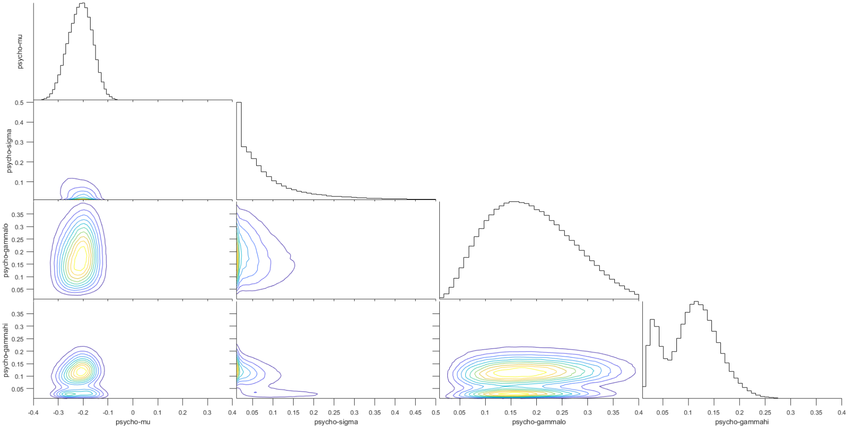
https://github.com/lacerbi/bads

Bayesian posteriors



$n = 1353$ trials

Bayesian posteriors



$n = 90$ trials

Benefits of Bayesian posteriors

- Check for parameter uncertainty, trade-offs, identifiability
  - Deeper understanding of your model
  - Robustness of claims (Acerbi, Ma, Vijayakumar, *NeurIPS* 2014)
- Less overfitting
- Use posterior samples to compute model comparison metrics
  - DIC, WAIC, LOO-CV
- Fully taking into account uncertainty is just *better*

How do I get Bayesian posteriors?

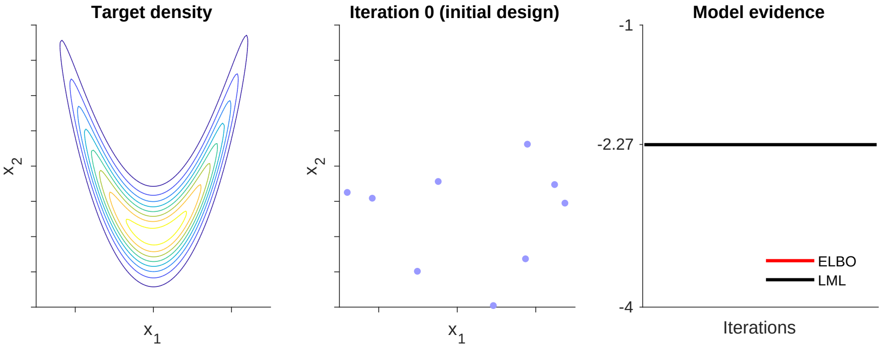
- Markov Chain Monte Carlo (MCMC; e.g. slice sampling, NUTS)
- Variational inference

Variational Bayesian Monte Carlo

Alternative to MCMC (for low-D, moderately costly problems)

Variational Bayesian Monte Carlo

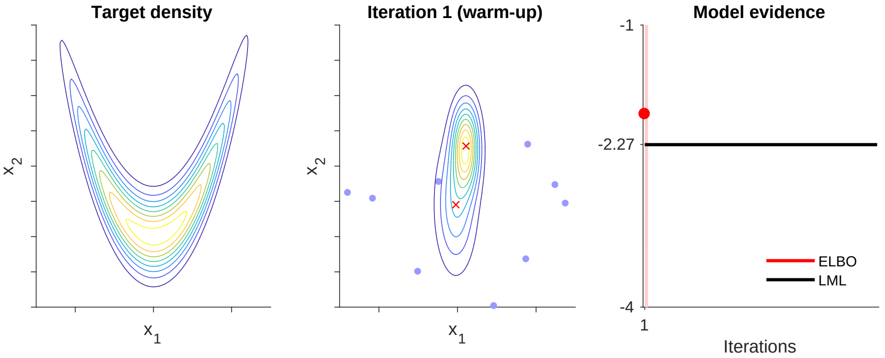
Alternative to MCMC (for low-D, moderately costly problems)



Acerbi, *NeurIPS* 2018

Variational Bayesian Monte Carlo

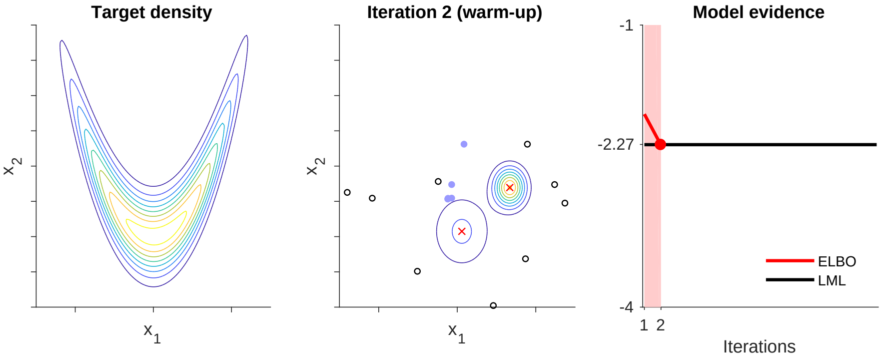
Alternative to MCMC (for low-D, moderately costly problems)



Acerbi, *NeurIPS* 2018

Variational Bayesian Monte Carlo

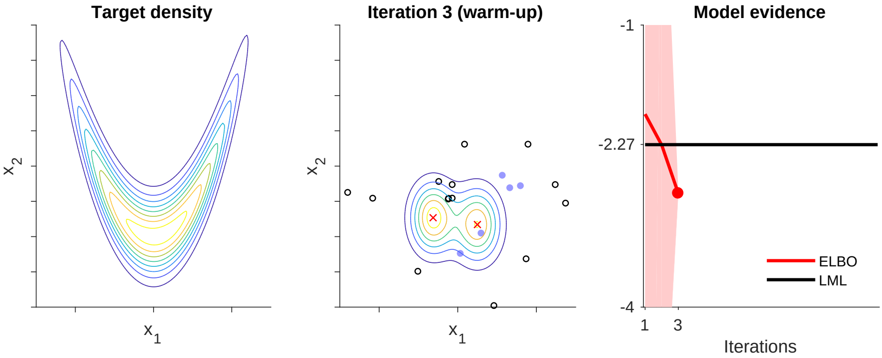
Alternative to MCMC (for low-D, moderately costly problems)



Acerbi, *NeurIPS* 2018

Variational Bayesian Monte Carlo

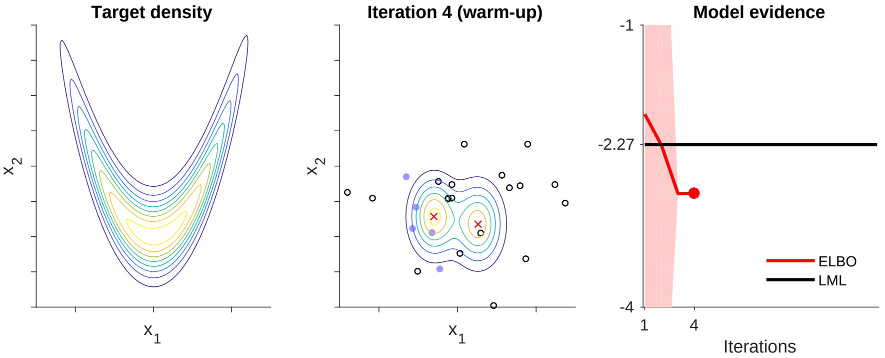
Alternative to MCMC (for low-D, moderately costly problems)



Acerbi, *NeurIPS* 2018

Variational Bayesian Monte Carlo

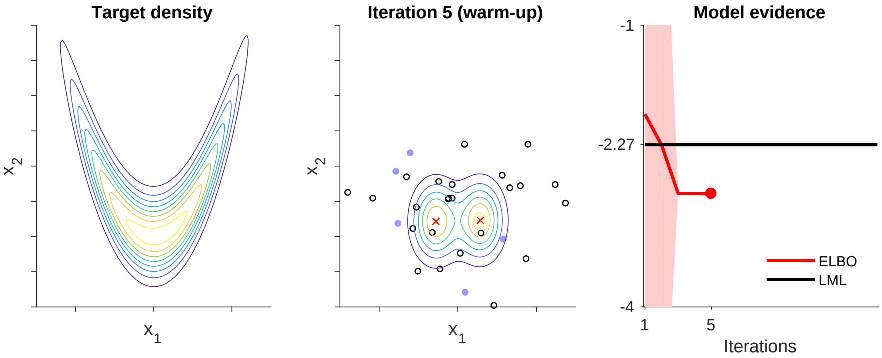
Alternative to MCMC (for low-D, moderately costly problems)



Acerbi, *NeurIPS* 2018

Variational Bayesian Monte Carlo

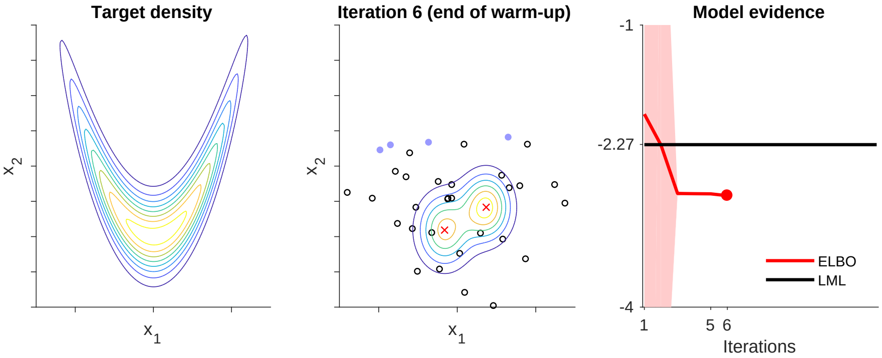
Alternative to MCMC (for low-D, moderately costly problems)



Acerbi, *NeurIPS* 2018

Variational Bayesian Monte Carlo

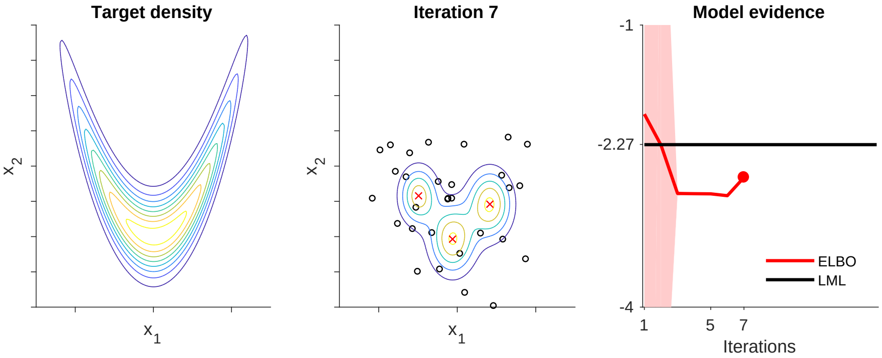
Alternative to MCMC (for low-D, moderately costly problems)



Acerbi, *NeurIPS* 2018

Variational Bayesian Monte Carlo

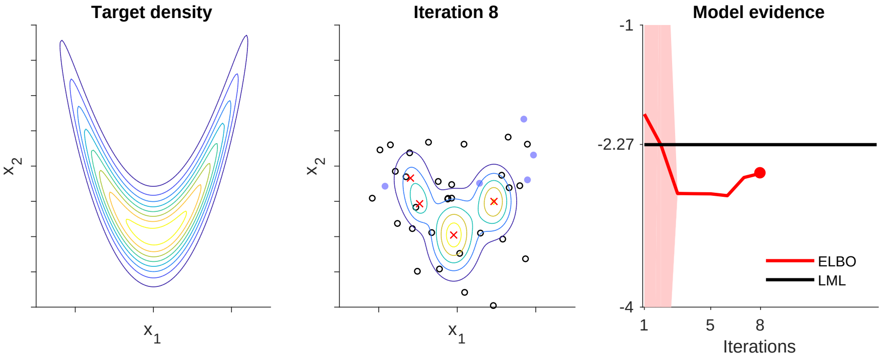
Alternative to MCMC (for low-D, moderately costly problems)



Acerbi, *NeurIPS* 2018

Variational Bayesian Monte Carlo

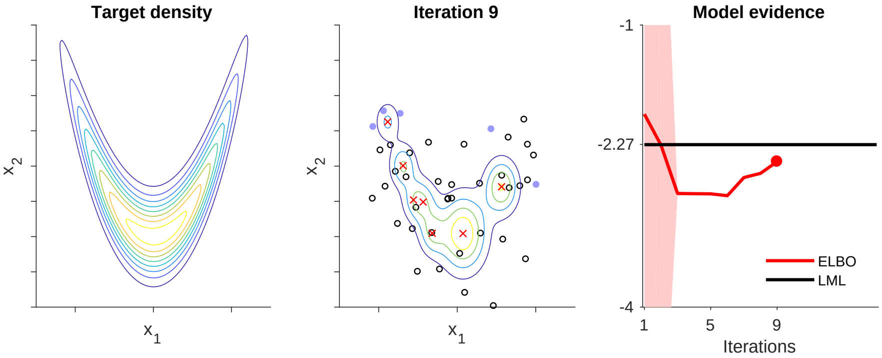
Alternative to MCMC (for low-D, moderately costly problems)



Acerbi, *NeurIPS* 2018

Variational Bayesian Monte Carlo

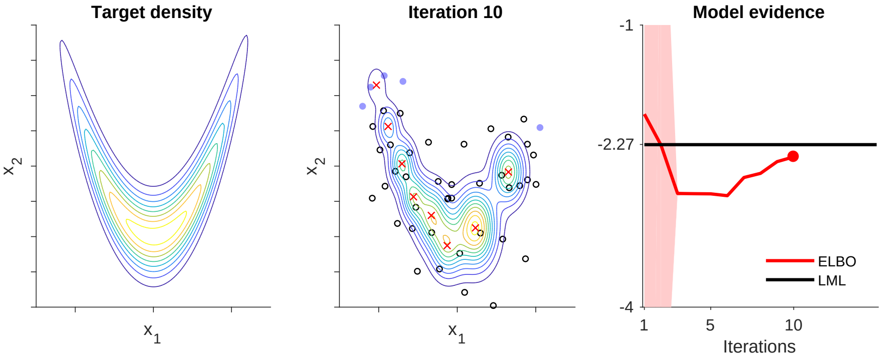
Alternative to MCMC (for low-D, moderately costly problems)



Acerbi, *NeurIPS* 2018

Variational Bayesian Monte Carlo

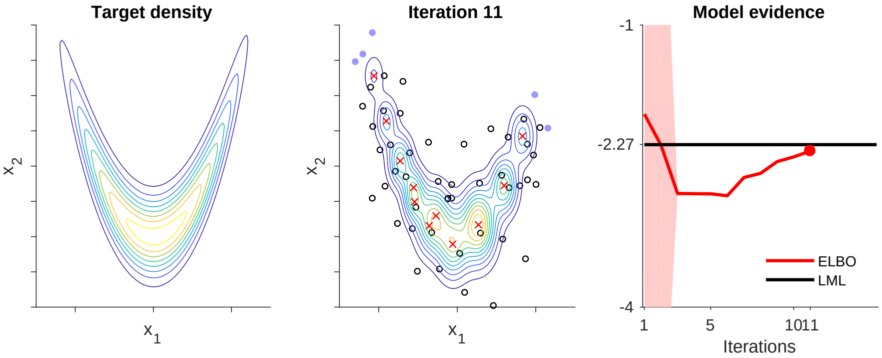
Alternative to MCMC (for low-D, moderately costly problems)



Acerbi, *NeurIPS* 2018

Variational Bayesian Monte Carlo

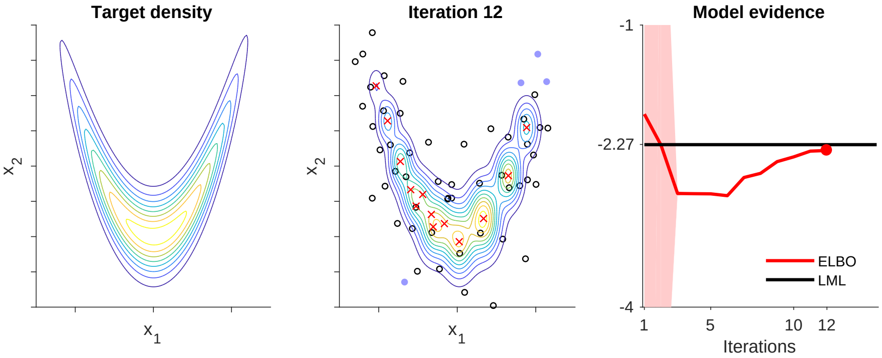
Alternative to MCMC (for low-D, moderately costly problems)



Acerbi, *NeurIPS* 2018

Variational Bayesian Monte Carlo

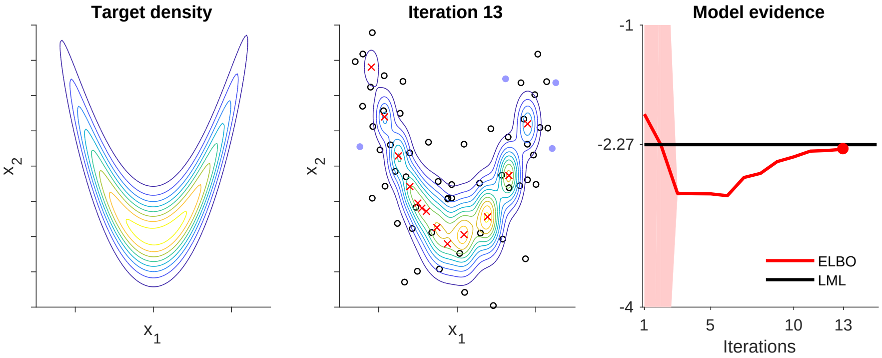
Alternative to MCMC (for low-D, moderately costly problems)



Acerbi, *NeurIPS* 2018

Variational Bayesian Monte Carlo

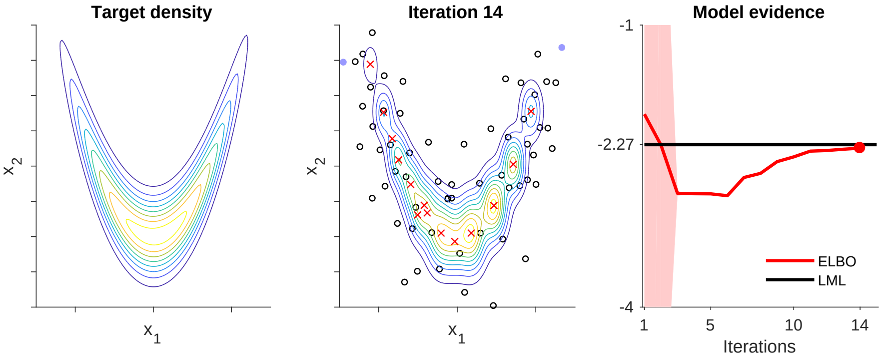
Alternative to MCMC (for low-D, moderately costly problems)



Acerbi, *NeurIPS* 2018

Variational Bayesian Monte Carlo

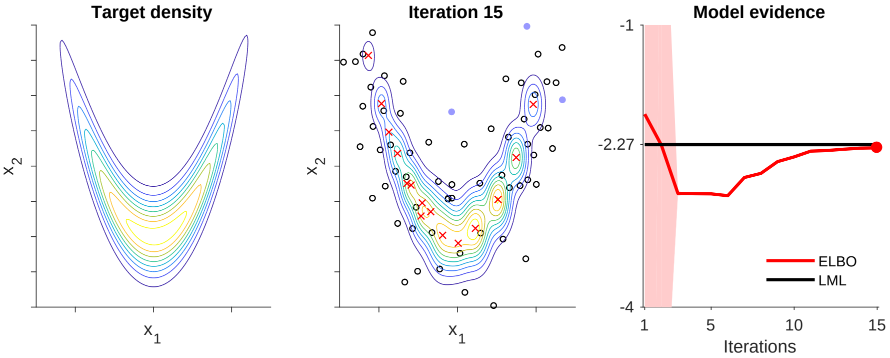
Alternative to MCMC (for low-D, moderately costly problems)



Acerbi, *NeurIPS* 2018

Variational Bayesian Monte Carlo

Alternative to MCMC (for low-D, moderately costly problems)



Acerbi, *NeurIPS* 2018

Applied example

OPEN ACCESS PEER-REVIEWED

RESEARCH ARTICLE

Bayesian comparison of explicit and implicit causal inference strategies in multisensory heading perception

Luigi Acerbi, Kalpana Dokka, Dora E. Angelaki, Wei Ji Ma

Published: July 27, 2018 • https://doi.org/10.1371/journal.pcbi.1006110

Final slide

- Contact me at `luigi.acerbi@unige.ch`
- Python tutorial: `github.com/lacerbi/tics-2020-tutorial`
- Optimization demos: `github.com/lacerbi/optimviz`

MATLAB toolboxes:

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

Thanks!

(Time for questions?)