Introduction to Modeling & Model Fitting for Psychophysics and Neuroscience

Luigi Acerbi

Department of Basic Neuroscience, University of Geneva
International Brain Lab

May, 2019

1. Introduction
   - Of models and likelihoods
2. Model fitting via optimization
   - An introduction to optimization
   - Optimization algorithms
   - Bayesian Optimization and BADS
3. Model selection via point estimates and little more
   - AIC/AICc
   - BIC
   - Cross-validation (CV)
   - Marginal likelihood and Laplace approximation
4. A couple of slides about MCMC

What is a model?

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

What is a mathematical model?

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

$$p\,(\mathrm{data}|\boldsymbol{\theta})$$

  - data is a dataset with n data points (e.g., trials)
  - $\boldsymbol{\theta}$ is a parameter vector

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

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

What is a mathematical model?

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

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

  - data is a dataset with n data points (e.g., trials)
  - $\boldsymbol{\theta}$ is a parameter vector

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

- Defining $p(\text{data}|\boldsymbol{\theta})$ is the core of model building

What is a mathematical model?

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

$$p\,(\mathrm{data}|\boldsymbol{\theta})$$

  - data is a dataset with n data points (e.g., trials)
  - $\boldsymbol{\theta}$ is a parameter vector

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

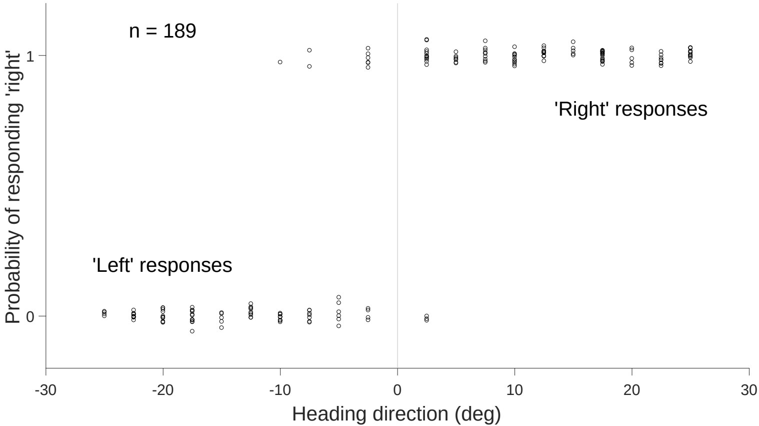
- Defining $p(\mathrm{data}|\boldsymbol{\theta})$ is the core of model building
  - Think about the data generation process!

Example: Psychometric function

Task: heading direction 'discrimination' task

Example: Psychometric function

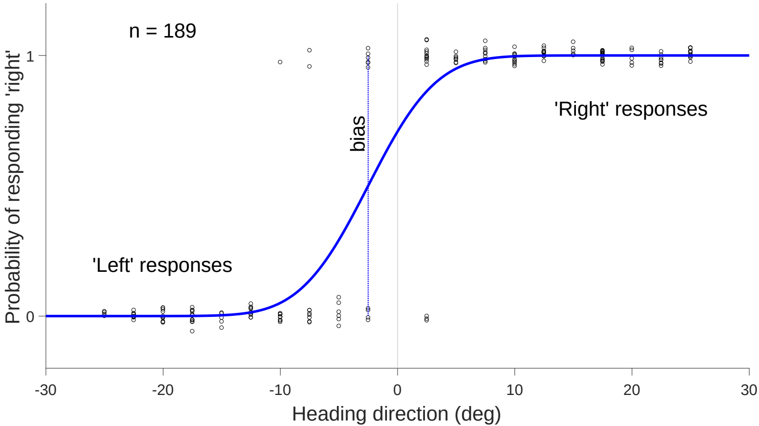
Task: heading direction 'discrimination' task



(data from Acerbi et al., *PLoS Comput Biol*, 2018)

Example: Psychometric function

Task: heading direction ‘discrimination’ task



(data from Acerbi et al., *PLoS Comput Biol*, 2018)

The log likelihood

- For numerical reasons we work with $\log p(\text{data}|\boldsymbol{\theta})$

The log likelihood

- For numerical reasons we work with $\log p(\text{data}|\boldsymbol{\theta})$
- When the data points are *conditionally independent*

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

The log likelihood

- For numerical reasons we work with $\log p(\text{data}|\boldsymbol{\theta})$
- When the data points are *conditionally independent*

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

- Write function that takes data and $\boldsymbol{\theta}$ as input arguments and returns $\log p(\text{data}|\boldsymbol{\theta})$

Model fitting

Model fitting $\sim$ *statistical estimation* problem

Model fitting

Model fitting $\sim$ *statistical estimation* problem

1. Maximum likelihood estimation (MLE)

Model fitting

Model fitting $\sim$ *statistical estimation* problem

1. Maximum likelihood estimation (MLE)

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

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

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

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

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

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

- For $n \to \infty$ converges to MLE (if $p(\hat{\boldsymbol{\theta}}_{\text{ML}}) \neq 0$)

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

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

- 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

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

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

- 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?

How to do model fitting?

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

- Model fitting $\sim$ *optimization problem*

How to do model fitting?

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

- Model fitting $\sim$ *optimization problem*

Approximate/full Bayesian posterior

- Things are a tad more complicated...
- Standard approach: Markov-Chain Monte Carlo (MCMC) sampling

1. Introduction
   - Of models and likelihoods

2. Model fitting via optimization
   - An introduction to optimization
   - Optimization algorithms
   - Bayesian Optimization and BADS

3. Model selection via point estimates and little more
   - AIC/AICc
   - BIC
   - Cross-validation (CV)
   - Marginal likelihood and Laplace approximation

4. A couple of slides about MCMC

The problem

- Given $f(\boldsymbol{x}) \equiv -\log p(\text{data}|\boldsymbol{x})$

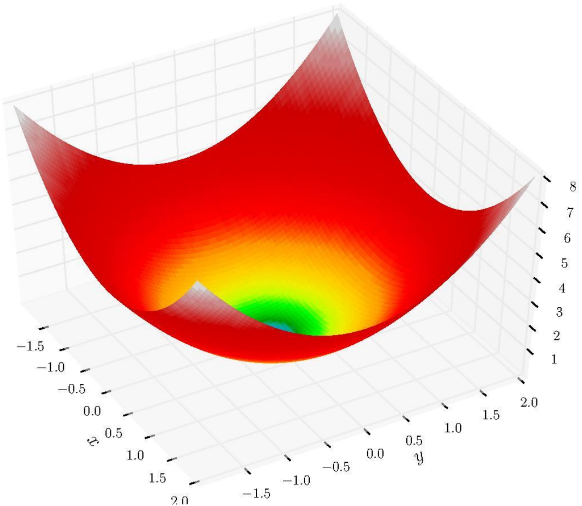
The problem

- Given $f(\boldsymbol{x}) \equiv -\log p(\text{data}|\boldsymbol{x})$
- Find $\boldsymbol{x}_{opt} \approx \arg\min_{\boldsymbol{x}} f(\boldsymbol{x})$ as fast as possible

The problem

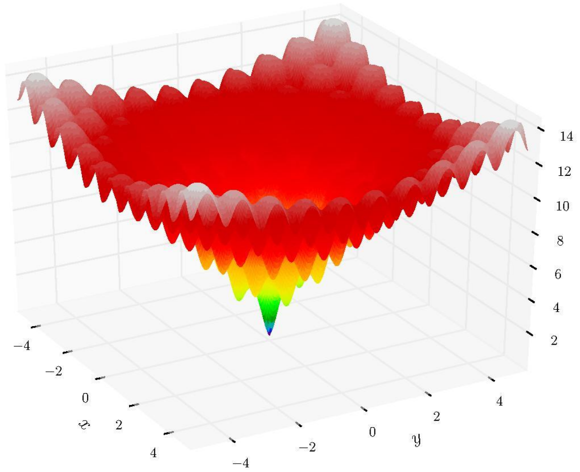
- Given $f(\boldsymbol{x}) \equiv -\log p(\text{data}|\boldsymbol{x})$
- Find $\boldsymbol{x}_{opt} \approx \arg\min_{\boldsymbol{x}} f(\boldsymbol{x})$ as fast as possible
- General case: $f(\boldsymbol{x})$ is a *black box*

How hard can it be?



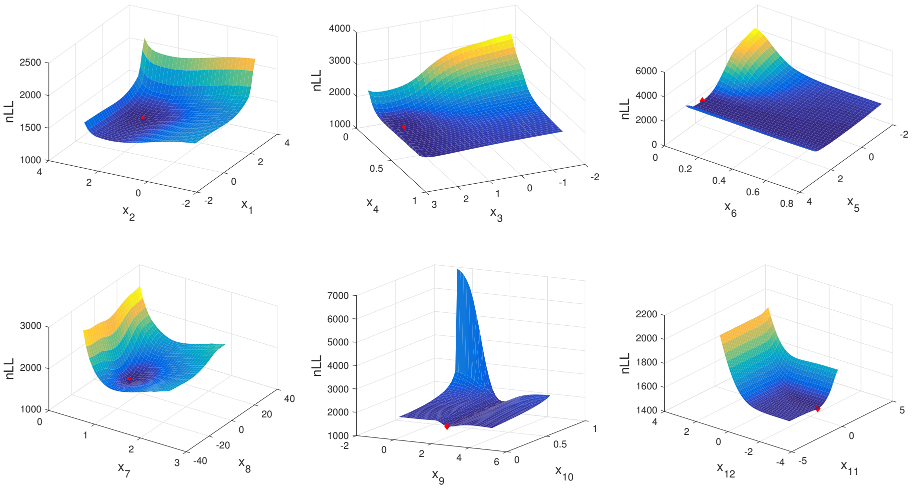
Source: Wikimedia Commons

How hard can it be?

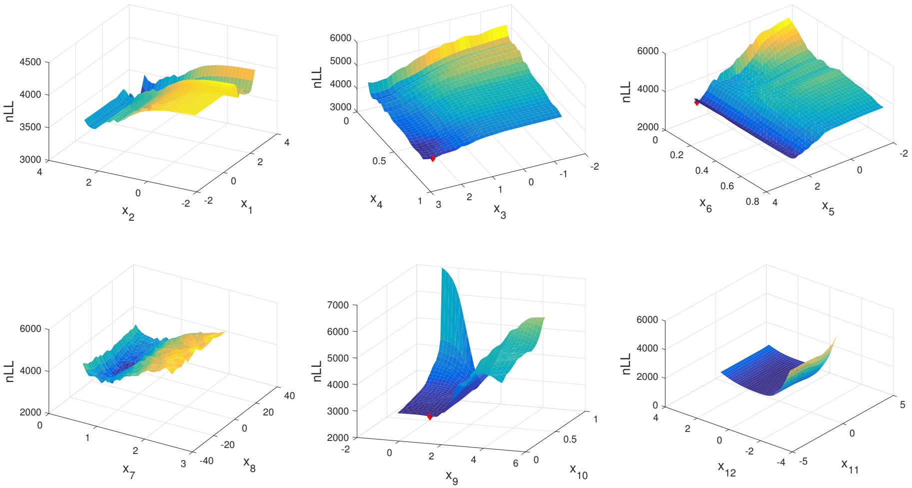


Source: Wikimedia Commons

How hard can it be?



How hard can it be?



How hard can it be?

neval	x_1	x_2	$f(\mathbf{x})$
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.)

Preparing for optimization

- *Domain* of parameter vector $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_k) \in \boldsymbol{\Theta}$

Preparing for optimization

- *Domain* of parameter vector $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_k) \in \boldsymbol{\Theta}$

  In practice, for each θ_k, define
  - The *hard bounds* of the parameter.
    - Mathematical constraints (e.g., $\sigma > 0$; $0 \leq p \leq 1$)
    - Effective physical limitations

Preparing for optimization

- *Domain* of parameter vector $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_k) \in \boldsymbol{\Theta}$

  In practice, for each θ_k, define
  - The *hard bounds* of the parameter.
    - Mathematical constraints (e.g., $\sigma > 0$; $0 \leq p \leq 1$)
    - Effective physical limitations
  - The *plausible bounds* of the parameter
    - Should span parameter values of all observers
    - Built from pilot studies, literature, guesswork
    - If in doubt, start larger

Preparing for optimization

- *Domain* of parameter vector $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_k) \in \boldsymbol{\Theta}$

  In practice, for each θ_k, define
  - The *hard bounds* of the parameter.
    - Mathematical constraints (e.g., $\sigma > 0$; $0 \leq p \leq 1$)
    - Effective physical limitations
  - The *plausible bounds* of the parameter
    - Should span parameter values of all observers
    - Built from pilot studies, literature, guesswork
    - If in doubt, start larger
- Consider reparameterizations to achieve
  - Uniformity of effects across parameter range
  - Independence between parameters

Which algorithm to use?

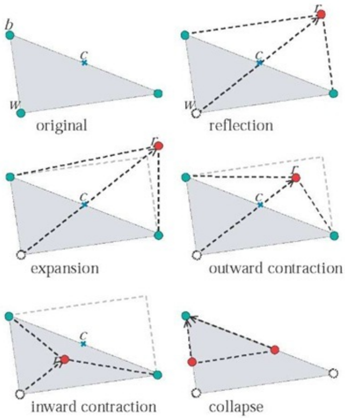
Deterministic		MATLAB Toolbox
Nelder-Mead	`fminsearch`	—
Quasi-Newton methods	`fminunc`,`fmincon`	Optimization
Direct search	`patternsearch`	Global Optimization
Multi-level Coordinate Search	`mcs`	— (free)
Stochastic		
Simulated Annealing	`simulannealbnd`	Global Optimization
Genetic Algorithm	`ga`	Global Optimization
Particle Swarm	`particleswarm`	Global Optimization
CMA-ES	`cmaes`	— (free)
Bayesian Optimization	`bayesopt`	Stats & ML
Bayesian Adaptive Direct Search	`bads`	— (free)

Which algorithm to use?

Deterministic		MATLAB Toolbox
Nelder-Mead	`fminsearch`	—
Quasi-Newton methods	`fminunc`,`fmincon`	Optimization
Direct search	`patternsearch`	Global Optimization
Multi-level Coordinate Search	`mcs`	— (free)
Stochastic		
Simulated Annealing	`simulannealbnd`	Global Optimization
Genetic Algorithm	`ga`	Global Optimization
Particle Swarm	`particleswarm`	Global Optimization
CMA-ES	`cmaes`	— (free)
Bayesian Optimization	`bayesopt`	Stats & ML
Bayesian Adaptive Direct Search	`bads`	— (free)

Nelder-Mead (`fminsearch`)

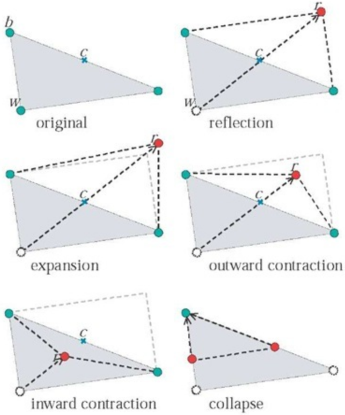
J. A. Nelder & R. Mead, A simplex method for function minimization (1965)



Source: Encyclopedia of Artificial Intelligence (2009)

Nelder-Mead (`fminsearch`)

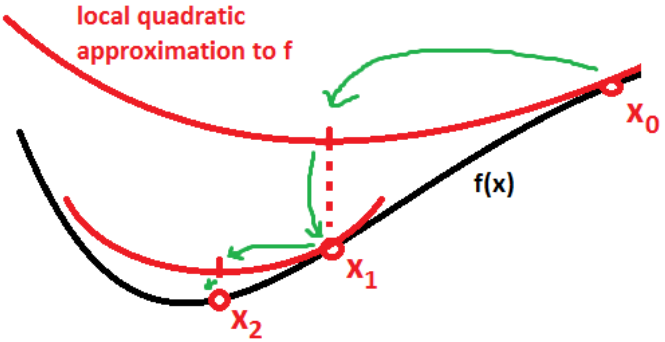
J. A. Nelder & R. Mead, A simplex method for function minimization (1965)



Source: Encyclopedia of Artificial Intelligence (2009)

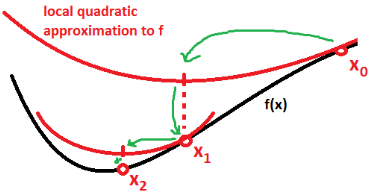
Bounded optimization: `fminsearchbnd` (John d'Errico)

Newton method



Source: StackExchange

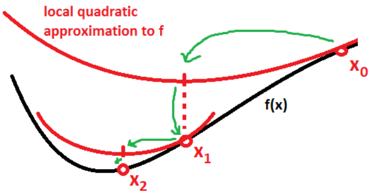
Newton method



Source: StackExchange

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

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

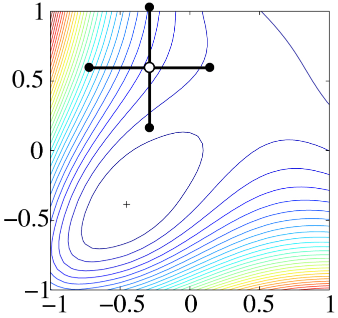


Source: StackExchange

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)

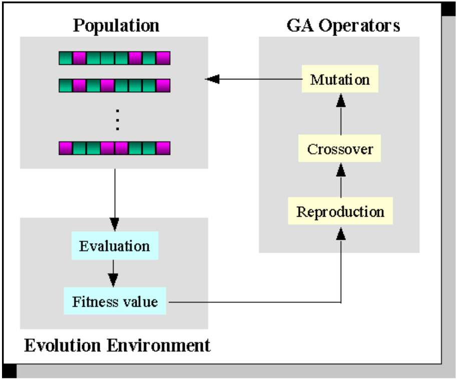
J.H. Holland, Adaptation in Natural and Artificial Systems (1975)

- Evolutionary algorithm
- Population based

Genetic Algorithms (ga)

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

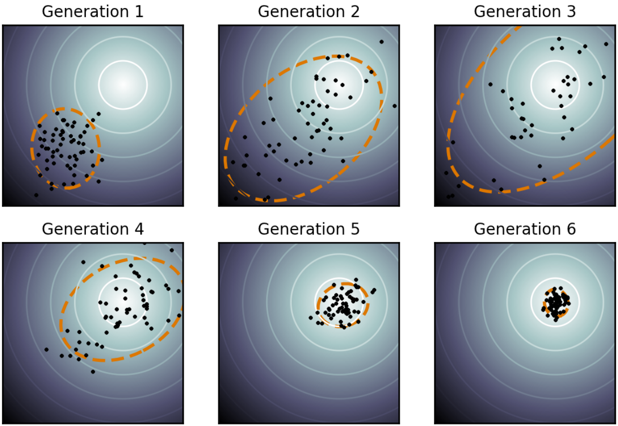
- Evolutionary algorithm
- Population based



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

Bayesian Optimization

J. Mockus, Application of Bayesian approach to numerical methods of global and stochastic optimization (1994)

Bayesian Optimization

J. Mockus, Application of Bayesian approach to numerical methods of global and stochastic optimization (1994)

1. Start with a prior over functions (Gaussian process)

Bayesian Optimization

J. Mockus, Application of Bayesian approach to numerical methods of global and stochastic optimization (1994)

1. Start with a prior over functions (Gaussian process)
2. Find $\tilde{\boldsymbol{x}}$ that maximizes acquisition function (exploration/exploitation)

Bayesian Optimization

J. Mockus, Application of Bayesian approach to numerical methods of global and stochastic optimization (1994)

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

Bayesian Optimization

J. Mockus, Application of Bayesian approach to numerical methods of global and stochastic optimization (1994)

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)

Bayesian Optimization

J. Mockus, Application of Bayesian approach to numerical methods of global and stochastic optimization (1994)

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`

Bayesian Optimization

- Good for expensive ($\gtrsim$ 10 mins), noisy functions up to $D \approx 20$

Bayesian Optimization

- Good for expensive ($\gtrsim 10$ mins), noisy functions up to $D \approx 20$
- Scales badly with n, computation time $\sim O(n^3)$

Bayesian Optimization

- Good for expensive ($\gtrsim 10$ mins), noisy functions up to $D \approx 20$
- Scales badly with n, computation time $\sim O(n^3)$
- Performance depends on quality of global approximation

Bayesian Adaptive Direct Search (`bads`)

- Combines Mesh-Adaptive Direct Search (MADS) with Bayesian Optimization (BO)

Bayesian Adaptive Direct Search (`bads`)

- Combines Mesh-Adaptive Direct Search (MADS) with Bayesian Optimization (BO)

Algorithm

1. Take as input f, `x0`, `LB`, `UB`, `PLB`, `PUB`
2. Evaluate f on an initial design and $x \leftarrow \arg\min_i f(x_i)$
3. Until convergence or `MaxFunEvals` do
    - Poll Step: Evaluate up to $2D$ points around x, update x
    - (Train Step: Train GP on neighborhood of x)
    - Search Step: Perform multiple iterations of BO in neighborhood of x

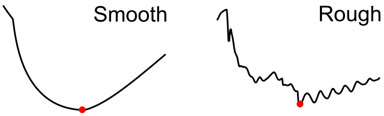
Acerbi & Ma, NeurIPS (2017)

Bayesian Adaptive Direct Search (`bads`)

- 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

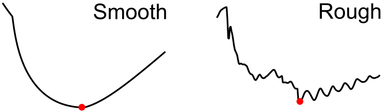
Optimization: The take-home slide

Optimization: The take-home slide



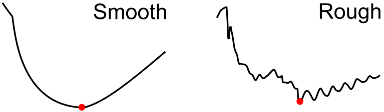
- Check your landscape

Optimization: The take-home slide



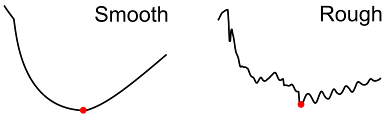
- Check your landscape
- If your problem is smooth $\Longrightarrow$ quasi-Newton (`fminunc`,`fmincon`)
  - If you can compute the gradient, do it!

Optimization: The take-home slide



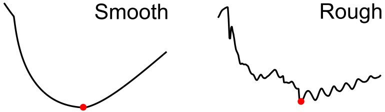
- Check your landscape
- If your problem is smooth $\Longrightarrow$ quasi-Newton (`fminunc`,`fmincon`)
  - If you can compute the gradient, do it!
- If your problem is rough or noisy...

Optimization: The take-home slide



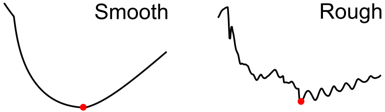
- Check your landscape
- If your problem is smooth $\Longrightarrow$ quasi-Newton (`fminunc`,`fmincon`)
  - If you can compute the gradient, do it!
- If your problem is rough or noisy...
  - Try to make it smooth and deterministic!

Optimization: The take-home slide

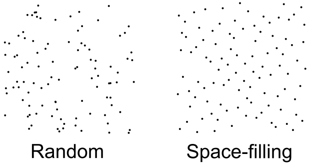


- Check your landscape
- If your problem is smooth $\Longrightarrow$ quasi-Newton (`fminunc`,`fmincon`)
  - If you can compute the gradient, do it!
- If your problem is rough or noisy...
  - Try to make it smooth and deterministic!
  - If the fcn is moderately costly $\Longrightarrow$ BADS

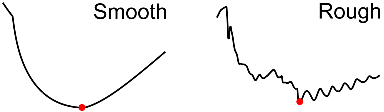
Optimization: The take-home slide



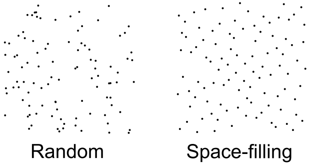
- Check your landscape
- If your problem is smooth $\Longrightarrow$ quasi-Newton (`fminunc`,`fmincon`)
  - If you can compute the gradient, do it!
- If your problem is rough or noisy. . .
  - Try to make it smooth and deterministic!
  - If the fcn is moderately costly $\Longrightarrow$ BADS
- Independently of the method, use several starting points
  - Use space-filling designs (Latin hypercubes, quasi-random sequences)



Optimization: The take-home slide



- Check your landscape
- If your problem is smooth $\Longrightarrow$ quasi-Newton (`fminunc`,`fmincon`)
  - If you can compute the gradient, do it!
- If your problem is rough or noisy...
  - Try to make it smooth and deterministic!
  - If the fcn is moderately costly $\Longrightarrow$ BADS
- Independently of the method, use several starting points
  - Use space-filling designs (Latin hypercubes, quasi-random sequences)



- If you can afford many fcn evals... consider MCMC instead of optimization!

1 Introduction

- Of models and likelihoods

2 Model fitting via optimization

- An introduction to optimization
- Optimization algorithms
- Bayesian Optimization and BADS

3 Model selection via point estimates and little more

- AIC/AICc
- BIC
- Cross-validation (CV)
- Marginal likelihood and Laplace approximation

4 A couple of slides about MCMC

The problem

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

The problem

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

Typical form of model comparison metric

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

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

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

- **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?

Akaike information criterion (AIC), part two

Why penalty is k?

(Do you really want to know?)

Akaike information criterion (AIC), part two

Why penalty is k?

Best predictive model

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

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* $KL(p_{\mathrm{true}} || p_m)$

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* $KL(p_{\mathrm{true}}||p_m)$
- $\left\langle \log p(y|\hat{\boldsymbol{\theta}}_{\mathrm{ML}}, \mathcal{M}_m) \right\rangle_{y \sim p_{\mathrm{true}}} \approx \frac{1}{n} \sum_{i=1}^{n} \log p(y_i|\hat{\boldsymbol{\theta}}_{\mathrm{ML}}, \mathcal{M}_m)$

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

Akaike information criterion (AIC), part two

Why penalty is k?

Best predictive model

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

- 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 $\approx \frac{1}{n} k$

Akaike information criterion (AIC), part two

Why penalty is k?

Best predictive model

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

- 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 $\boldsymbol{\theta} \longleftrightarrow p(y|\boldsymbol{\theta})$)

Corrected Akaike information criterion (AICc)

corrected Akaike information criterion

$$\text{AICc} = \log p(\text{data}|\hat{\boldsymbol{\theta}}_{\text{ML}}, \mathcal{M}_m) - k\left(\frac{n}{n-k-1}\right)$$

Corrected Akaike information criterion (AICc)

corrected Akaike information criterion

$$\text{AICc} = \log p(\text{data}|\hat{\theta}_{\text{ML}}, \mathcal{M}_m) - k\left(\frac{n}{n-k-1}\right)$$

- Correction derived for linear models
  - Still, better than AIC for small sample size

Schwarz (Bayesian) information criterion (BIC)

Bayesian information criterion

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

Schwarz (Bayesian) information criterion (BIC)

Bayesian information criterion

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

Schwarz (Bayesian) information criterion (BIC)

Bayesian information criterion

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

- **Goal:** Find true model
  - Assume $\mathcal{M}_{\text{true}}$ is in the model set
  - Based on loooose approximation of $P(\mathcal{M}|\text{data})$

Schwarz (Bayesian) information criterion (BIC)

Bayesian information criterion

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

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

Schwarz (Bayesian) information criterion (BIC)

Bayesian information criterion

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

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

Cross-validation

- **Goal:** Find best predictive model

Cross-validation

- **Goal:** Find best predictive model
  - Split data in training and validation

Cross-validation

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

Cross-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

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

Cross-validation

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

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

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

Cross-validation

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

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?

Marginal likelihood

Can we be more Bayesian?

(Not really, with only point estimates)

Marginal likelihood

Can we be more Bayesian?

- **Goal:** Find model with highest posterior probability

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

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

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

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

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

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:
  $P(\text{data}|\mathcal{M}) \approx \log p(\text{data}|\hat{\boldsymbol{\theta}}_{\text{ML}}, \mathcal{M}_m) + \frac{k}{2}\log 2\pi - \frac{1}{2}\log|\det \boldsymbol{H}(\boldsymbol{\theta}_{\text{ML}})|$

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

Model selection: The take-home slide

- AIC(c) vs BIC

Model selection: The take-home slide

- AIC(c) vs BIC
  - AIC(c) will often pick the most complex model

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

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

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

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

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

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

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

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)

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

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 recommended: 10-fold CV, or leave-one-out CV

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 recommended: 10-fold CV, or leave-one-out CV
  - Computationally expensive but might be worth it

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 recommended: 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

One slide about MCMC

One slide about MCMC

- Use MCMC

Another slide about MCMC

Another slide about MCMC

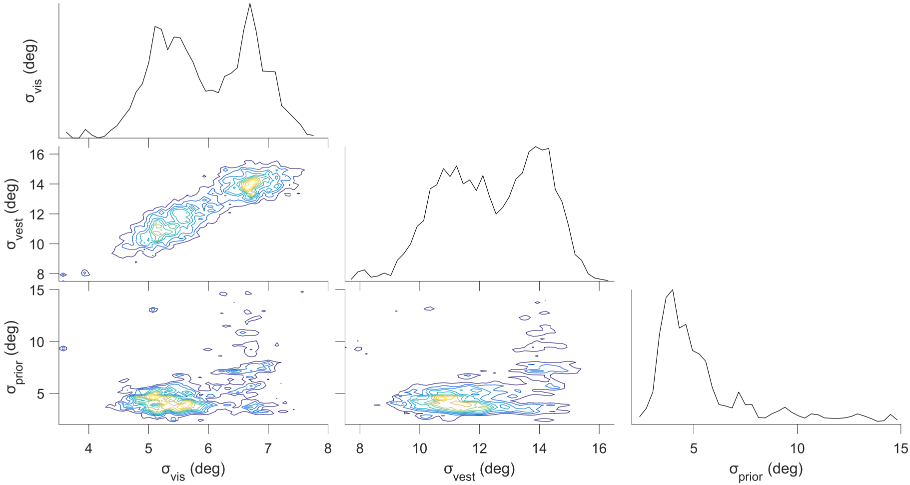


Figure made with `cornerplot.m`, by Will T. Adler

Another slide about MCMC

- Check for parameter uncertainty, trade-offs, identifiability
  - Deeper understanding of your model
  - Robustness of claims (Acerbi, Ma, Vijayakumar, 2014)

Another slide about MCMC

- Check for parameter uncertainty, trade-offs, identifiability
  - Deeper understanding of your model
  - Robustness of claims (Acerbi, Ma, Vijayakumar, 2014)
- Less overfitting

Another slide about MCMC

- Check for parameter uncertainty, trade-offs, identifiability
  - Deeper understanding of your model
  - Robustness of claims (Acerbi, Ma, Vijayakumar, 2014)
- Less overfitting
- Use posterior samples to compute model comparison metrics
  - DIC, WAIC, LOO-CV

Another slide about MCMC

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

Another slide about MCMC

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

But MCMC is finicky!

Another slide about MCMC

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

But MCMC is finicky!

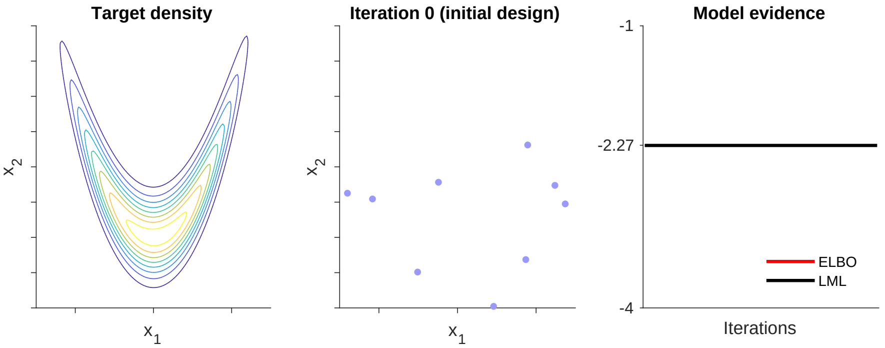
Use *slice sampling* (Neal, 2003)

Variational Bayesian Monte Carlo

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

Variational Bayesian Monte Carlo

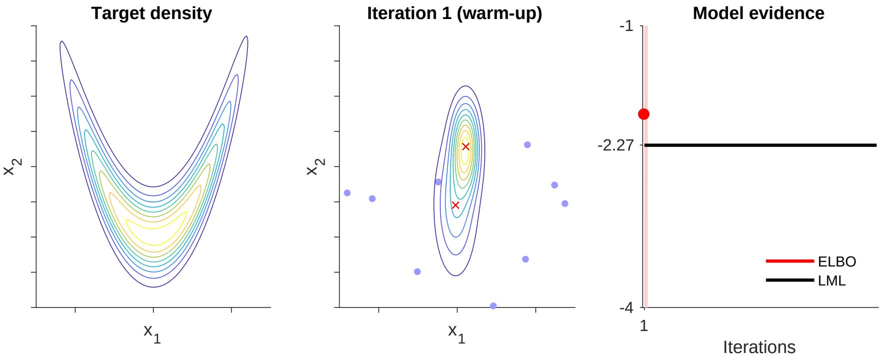
Alternative to MCMC (for low-D, moderately costly problems)



Acerbi, NeurIPS (2018)

Variational Bayesian Monte Carlo

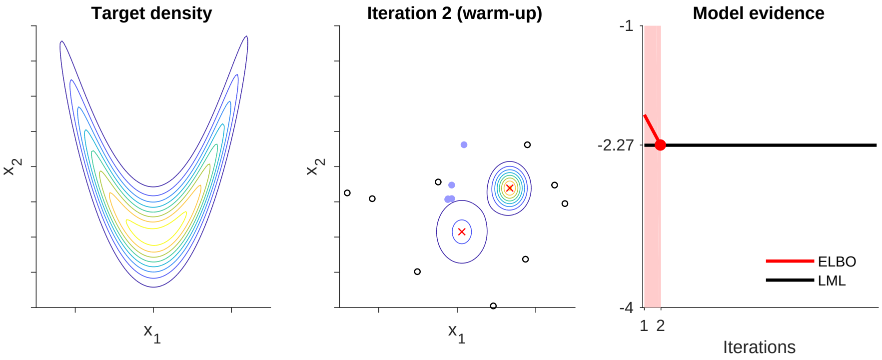
Alternative to MCMC (for low-D, moderately costly problems)



Acerbi, NeurIPS (2018)

Variational Bayesian Monte Carlo

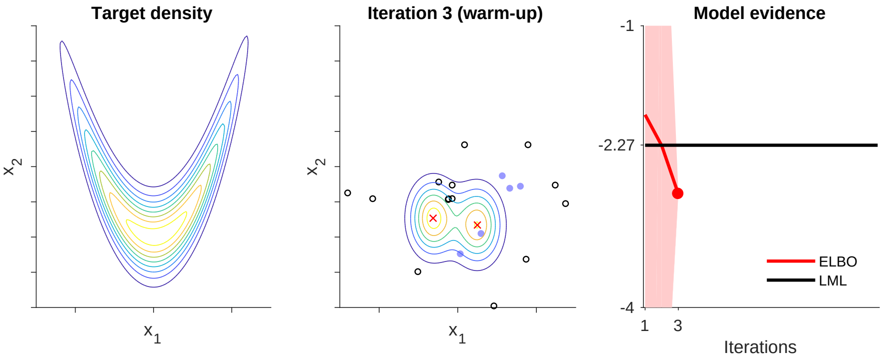
Alternative to MCMC (for low-D, moderately costly problems)



Acerbi, NeurIPS (2018)

Variational Bayesian Monte Carlo

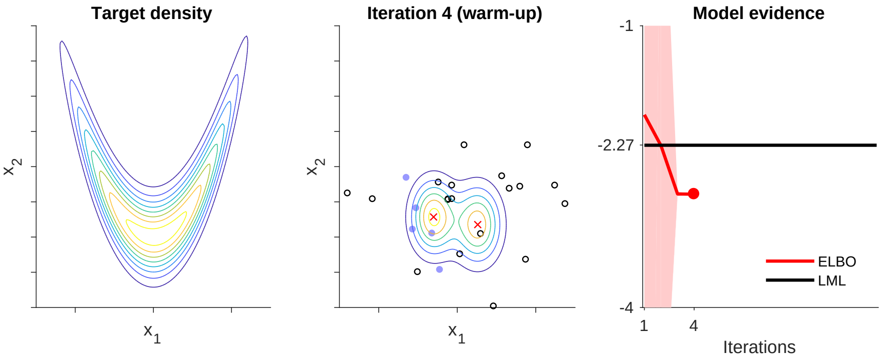
Alternative to MCMC (for low-D, moderately costly problems)



Acerbi, NeurIPS (2018)

Variational Bayesian Monte Carlo

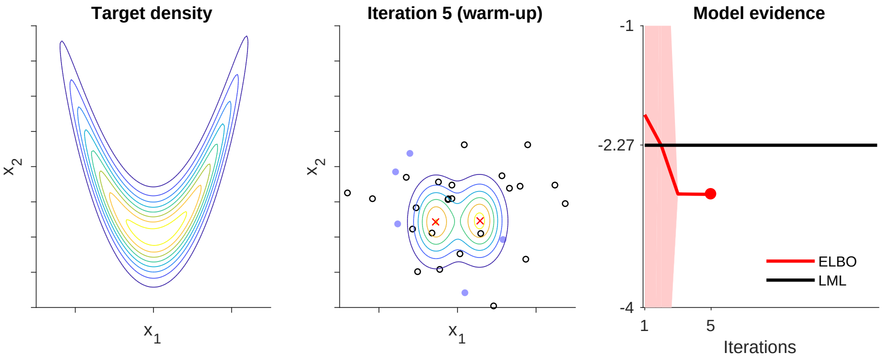
Alternative to MCMC (for low-D, moderately costly problems)



Acerbi, NeurIPS (2018)

Variational Bayesian Monte Carlo

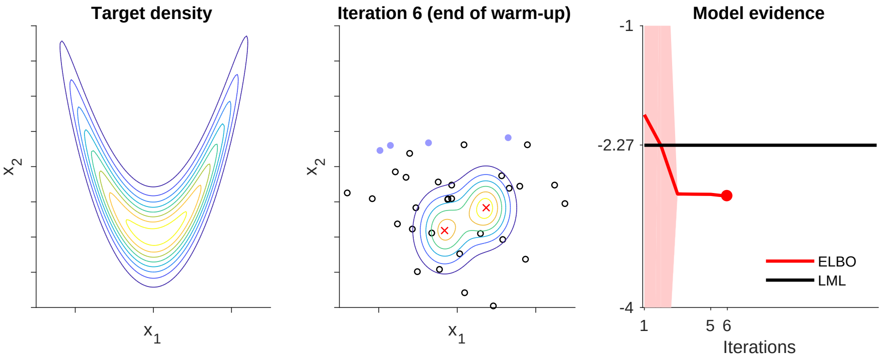
Alternative to MCMC (for low-D, moderately costly problems)



Acerbi, NeurIPS (2018)

Variational Bayesian Monte Carlo

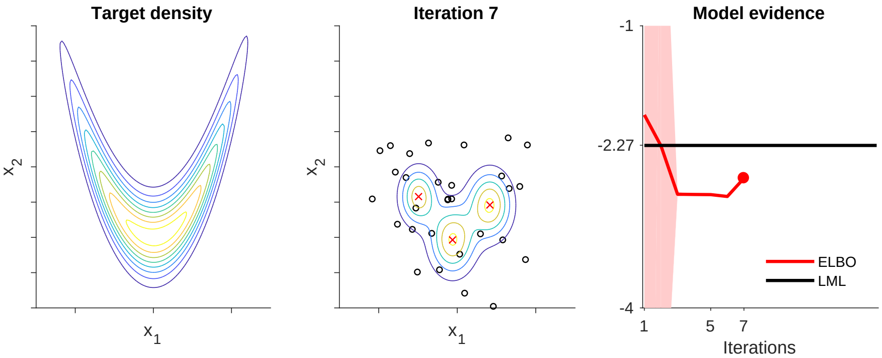
Alternative to MCMC (for low-D, moderately costly problems)



Acerbi, NeurIPS (2018)

Variational Bayesian Monte Carlo

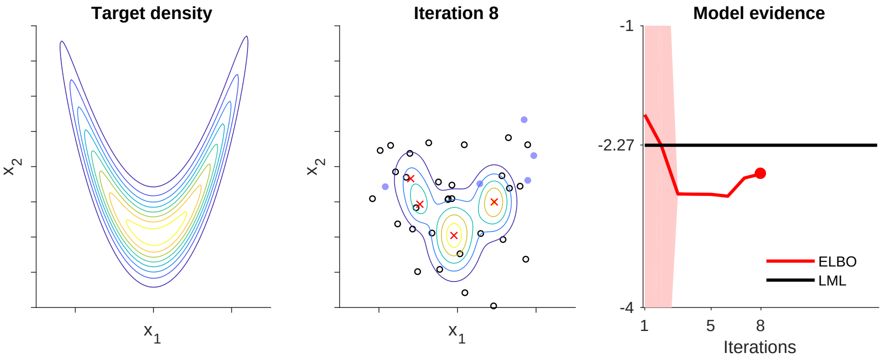
Alternative to MCMC (for low-D, moderately costly problems)



Acerbi, NeurIPS (2018)

Variational Bayesian Monte Carlo

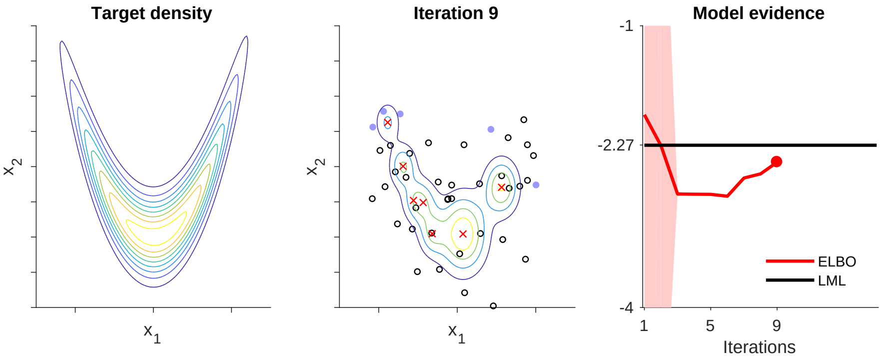
Alternative to MCMC (for low-D, moderately costly problems)



Acerbi, NeurIPS (2018)

Variational Bayesian Monte Carlo

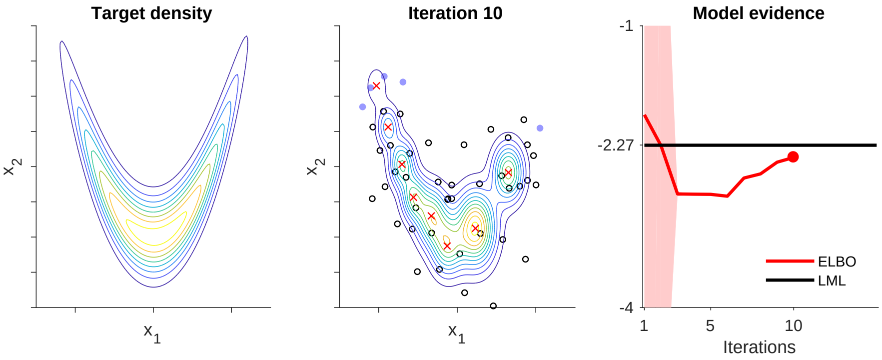
Alternative to MCMC (for low-D, moderately costly problems)



Acerbi, NeurIPS (2018)

Variational Bayesian Monte Carlo

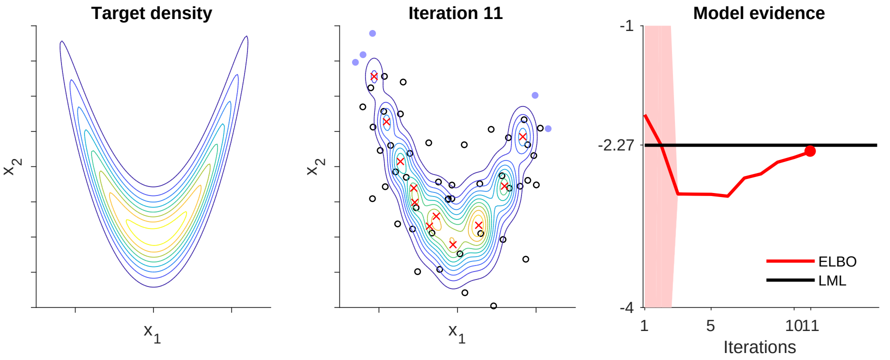
Alternative to MCMC (for low-D, moderately costly problems)



Acerbi, NeurIPS (2018)

Variational Bayesian Monte Carlo

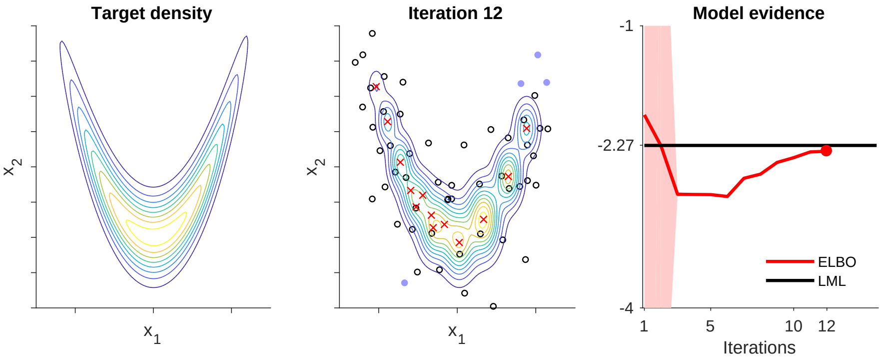
Alternative to MCMC (for low-D, moderately costly problems)



Acerbi, NeurIPS (2018)

Variational Bayesian Monte Carlo

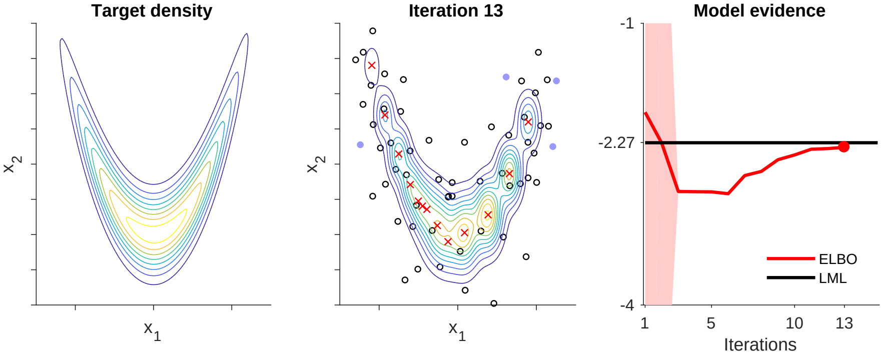
Alternative to MCMC (for low-D, moderately costly problems)



Acerbi, NeurIPS (2018)

Variational Bayesian Monte Carlo

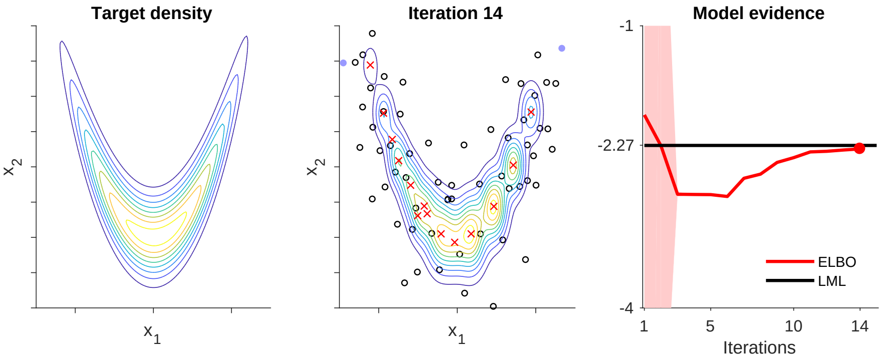
Alternative to MCMC (for low-D, moderately costly problems)



Acerbi, NeurIPS (2018)

Variational Bayesian Monte Carlo

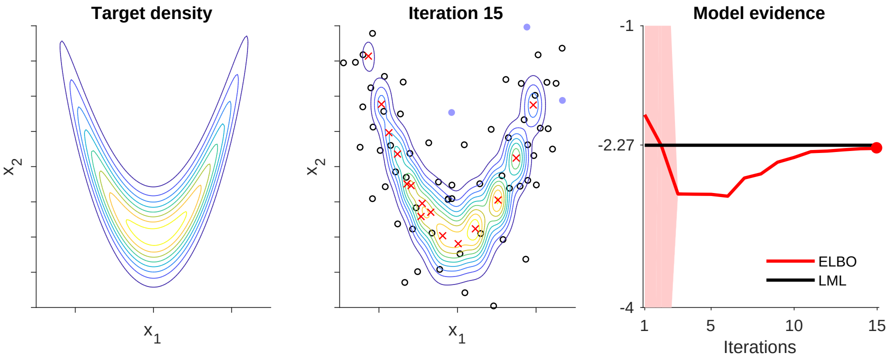
Alternative to MCMC (for low-D, moderately costly problems)



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@gmail.com
- Optimization demos: github.com/lacerbi/optimviz
- BADS available at github.com/lacerbi/bads
- VBMC available at github.com/lacerbi/vbmc
- Tutorial: github.com/lacerbi/workshop-bristol-2019

Final slide

- Contact me at luigi.acerbi@gmail.com
- Optimization demos: github.com/lacerbi/optimviz
- BADS available at github.com/lacerbi/bads
- VBMC available at github.com/lacerbi/vbmc
- Tutorial: github.com/lacerbi/workshop-bristol-2019

Thanks!

(Time for questions?)