Bayesian Regression with Functional Inequality Constraints

Joshua J. Bon with

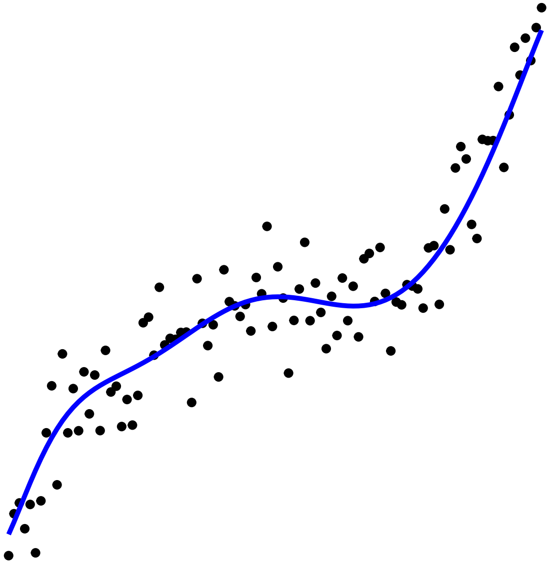
C. Drovandi, **K. Murray** & **B.A. Turlach**

30th of November, 2017

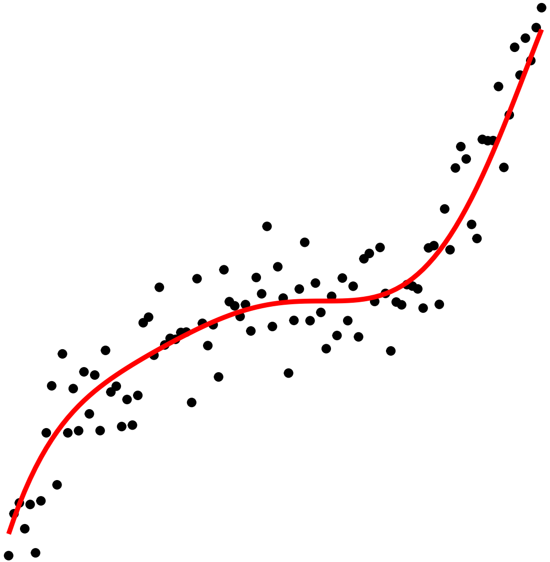
School of Mathematics and Statistics, University of Western Australia

An uphill battle

An uphill battle



An uphill battle



Functional constraints

Functional constraints: Inequality constraints defined by a function that varies over an auxiliary set.

$$\boldsymbol{\beta} \in \mathbb{R}^p \text{ such that}$$
$$f(\boldsymbol{x}; \boldsymbol{\beta}) \geq c, \forall \boldsymbol{x} \in X \subseteq \mathbb{R}^d$$

Most often in regression problems these arise as shape constraints over certain regions, e.g. monotonic polynomials.

Functional constraints

A parameter space with functional constraints takes the form

$$\Omega = \{\boldsymbol{\beta} \in \mathbb{R}^p : f(\boldsymbol{x}; \boldsymbol{\beta}) \geq c, \forall \boldsymbol{x} \in X\}$$

It is not feasible to check all $\boldsymbol{x} \in X$ (infinite points to check).

Instead, rewrite the constraint as

$$\Omega = \left\{\boldsymbol{\beta} \in \mathbb{R}^p : \min_{\boldsymbol{x} \in X} f(\boldsymbol{x}; \boldsymbol{\beta}) \geq c\right\}$$

Parameter constraints in Bayesian models

Incorporate information from the real world context into the prior probability of the parameters.

A constrained prior can be written as

$$\pi_c(\boldsymbol{\beta}) \propto \pi(\boldsymbol{\beta}) \times \mathbb{1}(\boldsymbol{\beta} \in \Omega)$$

$\boldsymbol{\beta} \notin \Omega$ are assigned a zero prior probability.

The posterior distribution is

$$\pi(\boldsymbol{\beta}|y) \propto \pi(y|\boldsymbol{\beta})\pi_c(\boldsymbol{\beta})$$

Computational considerations

When estimating parameters under functional constraints (Bayesian or ML) the computational difficulty is in assessing

$$\min_{\boldsymbol{x} \in X} f(\boldsymbol{x}; \boldsymbol{\beta}) \geq c$$

- When f is convex, a local minimum is the global minimum.
- Difficulties emerge when f is non-convex and $X \subseteq \mathbb{R}^d, d \geq 2$

Method: SMC + successively improving minima estimate

Sequential Monte Carlo

A subset of sequential Monte Carlo samplers[1] approximate the posterior distribution of static probabilistic models by:

- Evolving the starting distribution to the target distribution. For example, $\pi_t \propto \pi(y|\beta)^{\phi_t}\pi(\beta)$ with increasing $\phi_t \to 1$.
- In each iteration they attempt to sample from π_t by
  1. Weighted resampling from the previous particles $\sim \pi_{t-1}$
  2. Particle mutation to increase diversity (e.g. MCMC transition kernel with invariant distribution $\sim \pi_t$)

[1] P. Del Moral *et al.*, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **68**, 411–436 (2006).

Constrained Sequential Monte Carlo

Constrained sequential Monte Carlo samplers[2] move particles through a sequence of nested subsets until the constraint is satisfied. For our purposes we use:

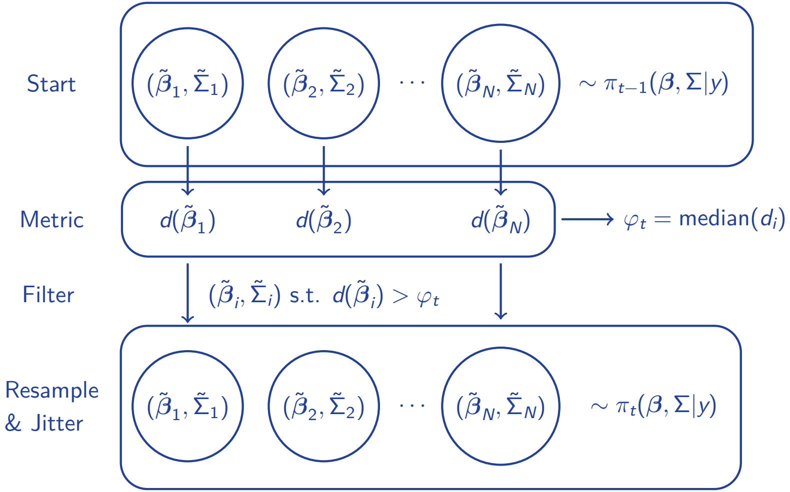
$$\pi_t(\boldsymbol{\beta}|y) \propto \pi(\boldsymbol{\beta}|y)\mathbb{1}(d(\boldsymbol{\beta}) > \varphi_t)$$

- $\pi(\boldsymbol{\beta}|y)$ is posterior distribution over the unconstrained space
- $d(\boldsymbol{\beta})$ measures "distance" away from the constrained space and $d(\boldsymbol{\beta}) \geq 0 \implies \boldsymbol{\beta} \in \Omega$
- $-\infty = \varphi_0 < \cdots < \varphi_t < \varphi_{t+1} < \cdots < \varphi_T = 0$
- Functional constraints: $d(\boldsymbol{\beta}) = \min_{\boldsymbol{x} \in X}\{f(\boldsymbol{x}; \boldsymbol{\beta}) - c\}$

[2] S. Golchi, D. A. Campbell, *Computational Statistics & Data Analysis* **97**, 98–113 (2016).

cSMC Algorithm Overview

1. Generate initial particles: $\{(\tilde{\boldsymbol{\beta}}_i, \tilde{\Sigma}_i)\}_{i=1}^{N} \sim \pi(\boldsymbol{\beta}, \Sigma | y)$
2. Calc metric: $d_i = d(\tilde{\boldsymbol{\beta}}_i)$, and $m = \sum_{i=1}^{N} \mathbb{1}(d_i \geq 0)$
3. Calc temp: $\varphi_t = \text{median}(\{d_i\}_{i=1}^{N})$ or $\varphi_t = 0$ if $m > N/2$
4. Resample N particles from $\{(\tilde{\boldsymbol{\beta}}_i, \tilde{\Sigma}_i)\}_{i=1}^{N}$ such that $d_i > \varphi_t$
5. Jitter/mutate kept particles with MCMC kernel $\sim \pi_t(\boldsymbol{\beta} | y)$
6. Repeat 2–5 until $m = N$, i.e. all $\boldsymbol{\beta} \in \Omega$



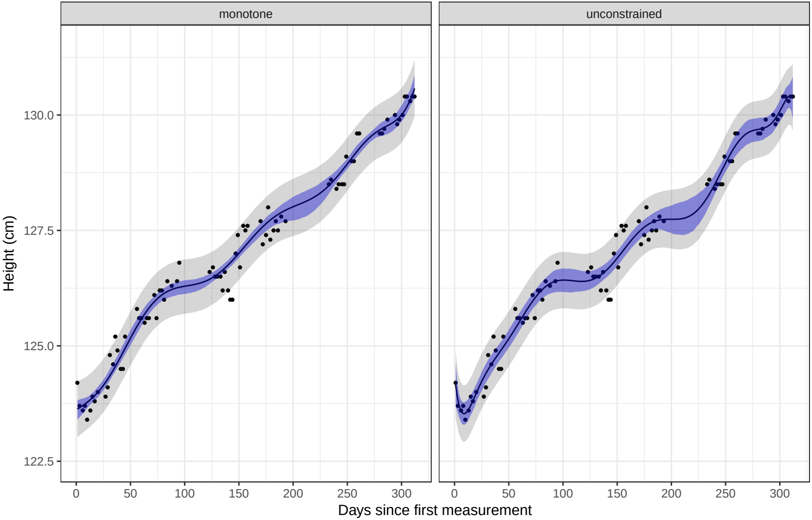
Start
$(\tilde{\boldsymbol{\beta}}_1, \tilde{\Sigma}_1)$
$(\tilde{\boldsymbol{\beta}}_2, \tilde{\Sigma}_2)$
$\cdots$
$(\tilde{\boldsymbol{\beta}}_N, \tilde{\Sigma}_N)$
$\sim \pi_{t-1}(\boldsymbol{\beta}, \Sigma | y)$
Metric
$d(\tilde{\boldsymbol{\beta}}_1)$
$d(\tilde{\boldsymbol{\beta}}_2)$
$d(\tilde{\boldsymbol{\beta}}_N)$
$\varphi_t = \text{median}(d_i)$
Filter
$(\tilde{\boldsymbol{\beta}}_i, \tilde{\Sigma}_i)$ s.t. $d(\tilde{\boldsymbol{\beta}}_i) > \varphi_t$
Resample
& Jitter
$(\tilde{\boldsymbol{\beta}}_1, \tilde{\Sigma}_1)$
$(\tilde{\boldsymbol{\beta}}_2, \tilde{\Sigma}_2)$
$\cdots$
$(\tilde{\boldsymbol{\beta}}_N, \tilde{\Sigma}_N)$
$\sim \pi_t(\boldsymbol{\beta}, \Sigma | y)$

Monotonicity

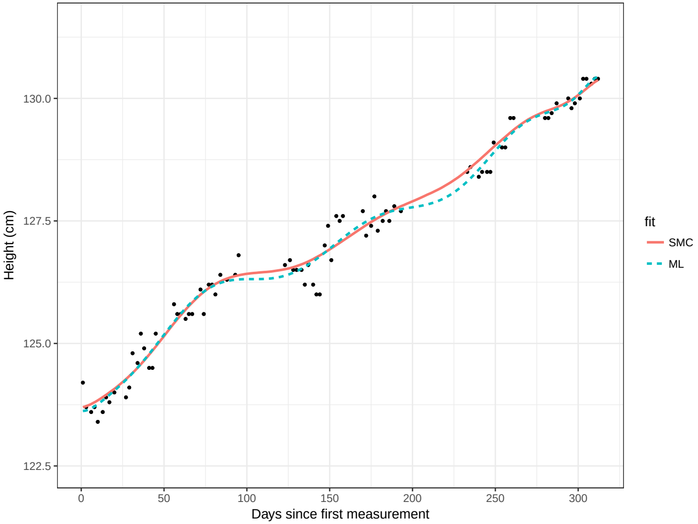
Monotonicity in one dimension:

$$d(\beta) = \min_{x \in [a,b]} f_x(x; \beta)$$

For twice-differentiable function, minimum can found at domain boundaries (if finite), or roots of $f_x(x; \beta)$.



Comparison of monotonic fit on “one child” dataset (Tuddenham and Snyder, 1954, `fda::onechild`). Fitted polynomials and 95% credible intervals for mean and posterior predictive distributions.



Maximum likelihood (ML) mean fitted values versus Bayesian mean fitted values (SMC). ML estimates from `MonoPoly` in `R`.

Monotonicity in two dimensions

Monotonicity in two dimensions:

$$d(\beta) = \min\left\{d_1(\beta), d_2(\beta)\right\} \text{ where } d_i(\beta) = \min_{\boldsymbol{x}\in X} f_{x_i}(\boldsymbol{x};\beta)$$

No guarantee that f_{x_1} or f_{x_2} is convex. When f is an arbitrary polynomial this is known to be a difficult problem (many local minima and 1-dimensional boundaries).

So we approximate the distance metric:

$$d_i(\beta) \approx \text{best local min of } f_{x_i}(\boldsymbol{x};\beta) \text{ from a set of starting points}$$

And attempt to improve the approximation each iteration

cSMC Method

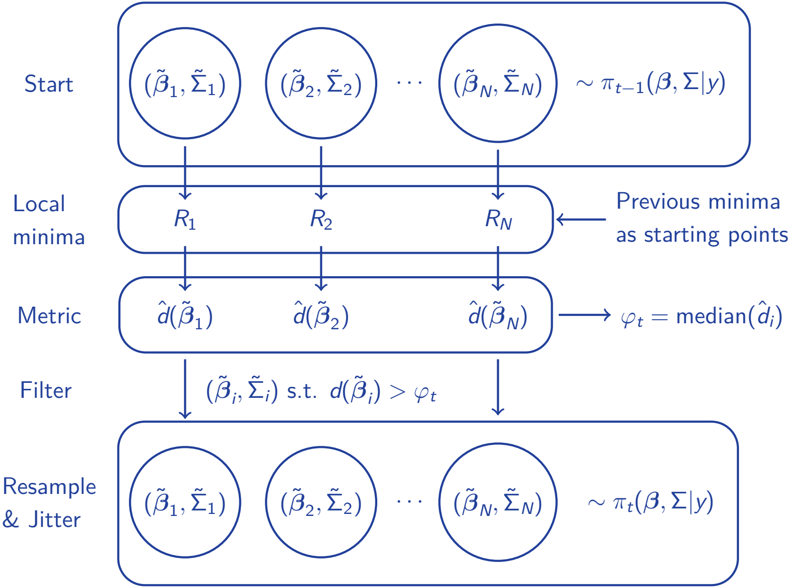
Particles now have 3 elements:

$$\{(\tilde{\boldsymbol{\beta}}_i, \tilde{\Sigma}_i, R_i)\}_{i=1}^{N}$$

where R_i are the local minima found. Every time a particle is mutated the R_i are updated with some optimisation routine using starting points:

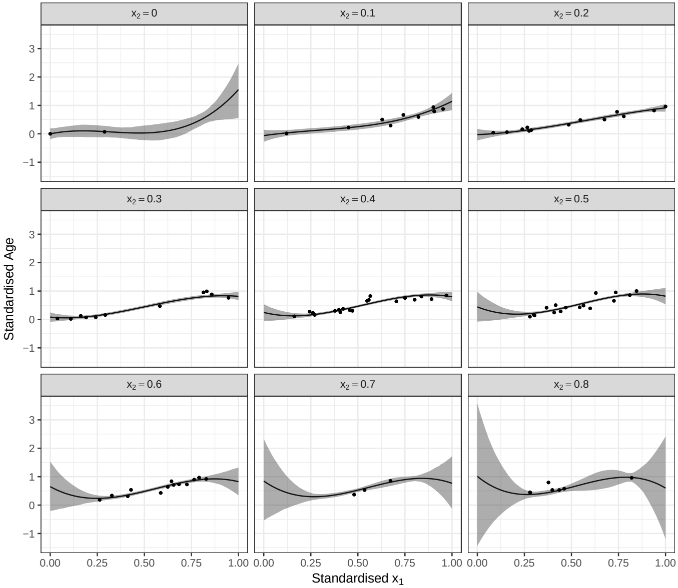
- The original R_i (likely to be close to new minima)
- Random sample of local minima across particles, i.e. $\{R_i\}_{i=1}^{N}$

Similar to a particle-swarm algorithm. Over time a global minima (if it exists) for each particle should be found.

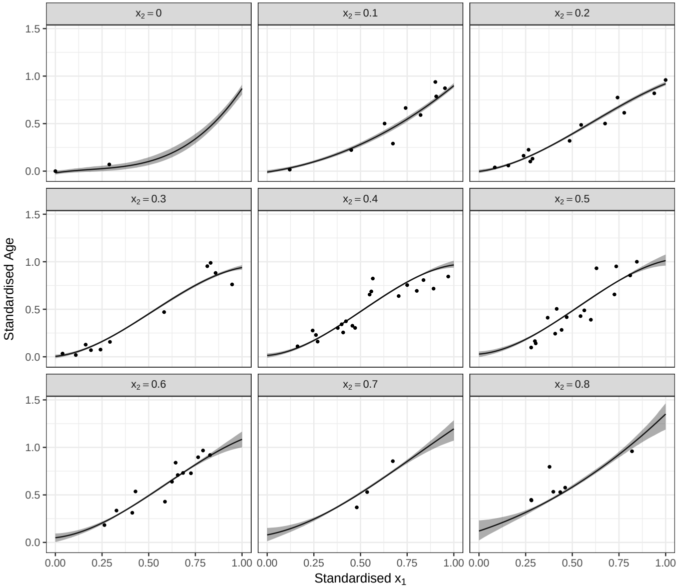

Start
$(\tilde{\beta}_1, \tilde{\Sigma}_1)$
$(\tilde{\beta}_2, \tilde{\Sigma}_2)$
$\cdots$
$(\tilde{\beta}_N, \tilde{\Sigma}_N)$
$\sim \pi_{t-1}(\beta, \Sigma | y)$
Local minima
R_1
R_2
R_N
Previous minima as starting points
Metric
$\hat{d}(\tilde{\beta}_1)$
$\hat{d}(\tilde{\beta}_2)$
$\hat{d}(\tilde{\beta}_N)$
$\varphi_t = \text{median}(\hat{d}_i)$
Filter
$(\tilde{\beta}_i, \tilde{\Sigma}_i)$ s.t. $d(\tilde{\beta}_i) > \varphi_t$
Resample & Jitter
$(\tilde{\beta}_1, \tilde{\Sigma}_1)$
$(\tilde{\beta}_2, \tilde{\Sigma}_2)$
$\cdots$
$(\tilde{\beta}_N, \tilde{\Sigma}_N)$
$\sim \pi_t(\beta, \Sigma | y)$

Data

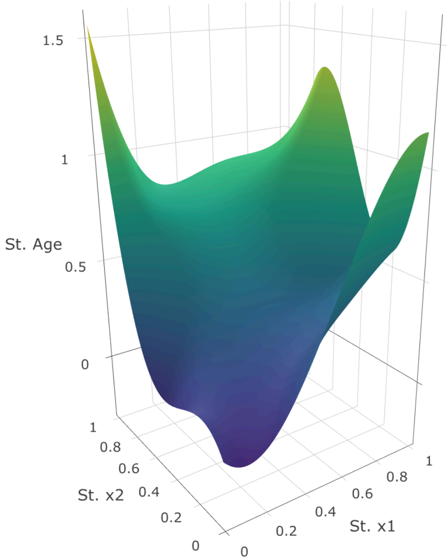
- Morphometric dataset
  - A number of morphometric measures taken from human skulls
  - Aged between 1 month and 19 years old
  - 174 individuals (93 male, 81 female)
- Aim: Predict age based on a selection of measurements
- In this example, using two measures that predict well individually for Male data.



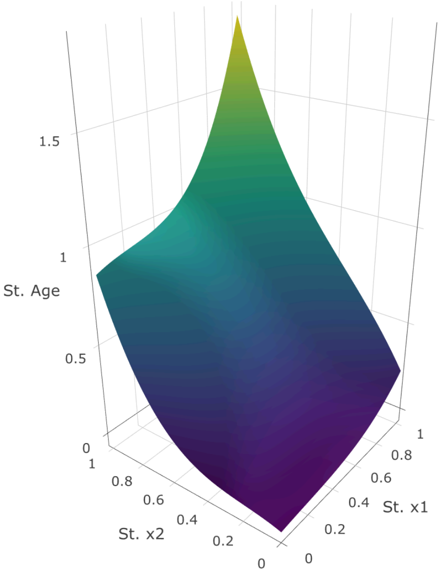
Unconstrained fit - 95% credible intervals for mean. Sliced by x_2.



Monotone fit - 95% credible intervals for mean. Sliced by x_2.



Unconstrained posterior mean fit.



Monotone posterior mean fit.

Conclusion

- Propose generic method for handling functional constraints in one or more dimensions
- Demonstrated with one- and two-dimensional monotonicity constraint
- What next?
  - Model selection
  - Incorporate measurement error techniques
  - Test on other datasets

Appendix

Comparison to maximum likelihood methods

ML regression generates an optimisation problem of the form

$$\max_{\boldsymbol{\beta}} \log\{\pi(y|\boldsymbol{\beta})\} \text{ s.t. } \boldsymbol{\beta} \in \Omega$$

again, where

$$\Omega = \{\boldsymbol{\beta} \in \mathbb{R}^p : f(\boldsymbol{x}; \boldsymbol{\beta}) \geq c, \forall \boldsymbol{x} \in X\}$$

This is a semi-infinite program, where X is often referred to as the index set.

Why Bayesian?

Due to the constraints, asymptotic ML theory is not applicable:

- Frequentist methods may need to rely on bootstrapping
- Whereas Bayesian computational methods can approximate entire posterior distribution

We can develop a Bayesian method to handle non-convex functional constraints with $d \geq 2$, based on:

- (Constrained) Sequential Monte Carlo
- Tracking local minima of the functional constraint

Other benefits

- No need to find optimal temperature for weights...