Parameter estimation via constraint propagation

Warwick Tucker

The CAPA group
Department of Mathematics
University of Uppsala, Sweden

UPPSALA
UNIVERSITET

Introduction

Problem formulation

A classic inverse problem/parameter estimation setting: given a finitely parametrized model function

$$y = f(x; p_1, p_2, \ldots, p_m) = f(x; p),$$

together with some (noisy) data

$$(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$$

and a search region $\mathcal{P}$ in parameter space, try to find parameters that give a *good agreement* between the data and the model.

UPPSALA UNIVERSITET

Introduction

Problem formulation

A classic inverse problem/parameter estimation setting: given a finitely parametrized model function

$$y = f(x; p_1, p_2, \ldots, p_m) = f(x; p),$$

together with some (noisy) data

$$(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$$

and a search region $\mathcal{P}$ in parameter space, try to find parameters that give a *good agreement* between the data and the model.

- Here, f can be almost anything (a function, an ODE, a PDE, some process...). This means that no single method is best.

Known difficulties

UPPSALA UNIVERSITET

Known difficulties

- **Existence:** with noisy data, or with an incorrect model, there is usually no parameter *at all* that produces a perfect fit between the model and the data.

Known difficulties

- **Existence:** with noisy data, or with an incorrect model, there is usually no parameter *at all* that produces a perfect fit between the model and the data.
- **Uniqueness:** even with *unlimited* amounts of *exact* data, there might not exist a unique solution $p^\sharp \in \mathcal{P}$ such that

$$f(x_i; p^\sharp) = y_i \qquad i = 1, \ldots, N.$$

UPPSALA UNIVERSITET

Known difficulties

- **Existence:** with noisy data, or with an incorrect model, there is usually no parameter *at all* that produces a perfect fit between the model and the data.
- **Uniqueness:** even with *unlimited* amounts of *exact* data, there might not exist a unique solution $p^\sharp \in \mathcal{P}$ such that

$$f(x_i; p^\sharp) = y_i \qquad i = 1, \ldots, N.$$

- **Instability:** many inverse problems are extremely unstable (ill-conditioned): a small perturbation in data produces a large change in the fitted parameter.

Introduction

A statistical approach

Assume that the model is correct, and that the data is perturbed via some probability distribution (almost always *normal*):

Introduction

A statistical approach

Assume that the model is correct, and that the data is perturbed via some probability distribution (almost always *normal*):

(1) **Generate data:** $y_i^\sharp = f(x_i; p^\sharp) \qquad i = 1, \ldots, N$

Introduction

A statistical approach

Assume that the model is correct, and that the data is perturbed via some probability distribution (almost always *normal*):

(1) **Generate data:** $y_i^\sharp = f(x_i; p^\sharp) \qquad i = 1, \ldots, N$

(2) **Perturb data:** $y_i = y_i^\sharp + \eta_i \qquad \eta_i \sim N(0, \sigma^2)$

A statistical approach

Assume that the model is correct, and that the data is perturbed via some probability distribution (almost always *normal*):

(1) **Generate data:** $y_i^\sharp = f(x_i; p^\sharp) \qquad i = 1, \ldots, N$

(2) **Perturb data:** $y_i = y_i^\sharp + \eta_i \qquad \eta_i \sim N(0, \sigma^2)$

Then use a (weighted) least-squares approach to find the best parameter.

Introduction

A statistical approach

Assume that the model is correct, and that the data is perturbed via some probability distribution (almost always *normal*):

$$
\begin{aligned}
&(1)\ \textbf{Generate data:}\quad y_i^\sharp = f(x_i;p^\sharp) && i = 1,\dots,N\\
&(2)\ \textbf{Perturb data:}\quad y_i = y_i^\sharp + \eta_i && \eta_i \sim N(0,\sigma^2)
\end{aligned}
$$

Then use a (weighted) least-squares approach to find the best parameter.

- If the parameters enter f linearly, this is "straight-forward".

UPPSALA UNIVERSITET

Introduction

A statistical approach

Assume that the model is correct, and that the data is perturbed via some probability distribution (almost always *normal*):

(1) **Generate data:** $y_i^\sharp = f(x_i; p^\sharp) \qquad i = 1, \ldots, N$

(2) **Perturb data:** $y_i = y_i^\sharp + \eta_i \qquad \eta_i \sim N(0, \sigma^2)$

Then use a (weighted) least-squares approach to find the best parameter.

- If the parameters enter f linearly, this is "straight-forward".
- Otherwise, we have moved the problem to *global optimization*.

Introduction

A statistical approach

Assume that the model is correct, and that the data is perturbed via some probability distribution (almost always *normal*):

$$\begin{aligned} &(1)\ \textbf{Generate data:} && y_i^\sharp = f(x_i; p^\sharp) && i = 1, \ldots, N \\ &(2)\ \textbf{Perturb data:} && y_i = y_i^\sharp + \eta_i && \eta_i \sim N(0, \sigma^2) \end{aligned}$$

Then use a (weighted) least-squares approach to find the best parameter.

- If the parameters enter f linearly, this is "straight-forward".
- Otherwise, we have moved the problem to *global optimization*.
- The selection of weights is almost always a delicate issue.

UPPSALA UNIVERSITET

A set-valued approach

Assume that the model is uncertain, and that the data is perturbed via some unknown mechanism.

Rather than seaching for the least-squares best fit, we attempt to locate nearby models that are *consistent* with nearby data:

A set-valued approach

Assume that the model is uncertain, and that the data is perturbed via some unknown mechanism.
Rather than seaching for the least-squares best fit, we attempt to locate nearby models that are *consistent* with nearby data:

$$\text{(1) } \textbf{Widen data: } \boldsymbol{y}_i = y_i(1 + \alpha[-1, 1]) \qquad i = 1, \ldots, N$$

Introduction

A set-valued approach

Assume that the model is uncertain, and that the data is perturbed via some unknown mechanism.
Rather than seaching for the least-squares best fit, we attempt to locate nearby models that are *consistent* with nearby data:

(1) **Widen data:** $\boldsymbol{y}_i = y_i(1 + \alpha[-1, 1]) \qquad i = 1, \ldots, N$

(2) **Consistent set:** $\mathcal{S} = \cap_{i=1}^{N}\{p \in \mathcal{P} \colon f(x_i; p) \in \boldsymbol{y}_i\}$.

UPPSALA UNIVERSITET

Introduction

A set-valued approach

Assume that the model is uncertain, and that the data is perturbed via some unknown mechanism.
Rather than seaching for the least-squares best fit, we attempt to locate nearby models that are *consistent* with nearby data:

(1) **Widen data:** $\boldsymbol{y}_i = y_i(1+\alpha[-1,1]) \qquad i = 1,\dots,N$
(2) **Consistent set:** $\mathcal{S} = \cap_{i=1}^{N}\{p \in \mathcal{P}\colon f(x_i;p) \in \boldsymbol{y}_i\}$.

Of course, $\mathcal{S}$ is very hard to find, but by discretizing the search space $\mathcal{P} \to \mathcal{P}_K$, we can form an inner/outer enclosure of $\mathcal{S}$:

$$\underline{\mathcal{S}} = \{\boldsymbol{p} \subset \mathcal{P}_K\colon f(x_i;\boldsymbol{p}) \subset \boldsymbol{y}_i \text{ for all } i = 1,\dots,N\}$$
$$\overline{\mathcal{S}} = \{\boldsymbol{p} \subset \mathcal{P}_K\colon f(x_i;\boldsymbol{p}) \cap \boldsymbol{y}_i \neq \emptyset \text{ for all } i = 1,\dots,N\}$$

UPPSALA UNIVERSITET

Introduction

A set-valued approach

Assume that the model is uncertain, and that the data is perturbed via some unknown mechanism.
Rather than seaching for the least-squares best fit, we attempt to locate nearby models that are *consistent* with nearby data:

(1) **Widen data:** $\boldsymbol{y}_i = y_i(1 + \alpha[-1, 1]) \qquad i = 1, \dots, N$

(2) **Consistent set:** $\mathcal{S} = \cap_{i=1}^{N} \{p \in \mathcal{P} \colon f(x_i; p) \in \boldsymbol{y}_i\}$.

Of course, $\mathcal{S}$ is very hard to find, but by discretizing the search space $\mathcal{P} \to \mathcal{P}_K$, we can form an inner/outer enclosure of $\mathcal{S}$:

$$\begin{aligned} \underline{\mathcal{S}} &= \{\boldsymbol{p} \subset \mathcal{P}_K \colon f(x_i; \boldsymbol{p}) \subset \boldsymbol{y}_i \text{ for all } i = 1, \dots, N\} \\ \overline{\mathcal{S}} &= \{\boldsymbol{p} \subset \mathcal{P}_K \colon f(x_i; \boldsymbol{p}) \cap \boldsymbol{y}_i \neq \emptyset \text{ for all } i = 1, \dots, N\} \end{aligned}$$

The coarser the discretization of $\mathcal{P}$, the less we trust the model.

UPPSALA UNIVERSITET

Set-valued computations

Interval analysis

All our computations are set-valued, and are based on the *inclusion principle*:

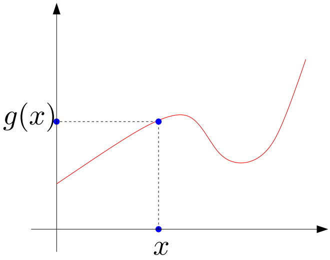
$$\mathrm{range}(g;\boldsymbol{x}) = \{g(x)\colon x \in \boldsymbol{x}\} \subseteq g(\boldsymbol{x})$$

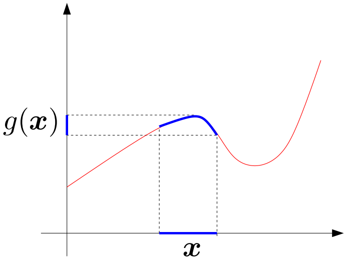
Set-valued computations

Interval analysis

All our computations are set-valued, and are based on the *inclusion principle*:

$$\mathrm{range}(g; \boldsymbol{x}) = \{g(x) \colon x \in \boldsymbol{x}\} \subseteq g(\boldsymbol{x})$$





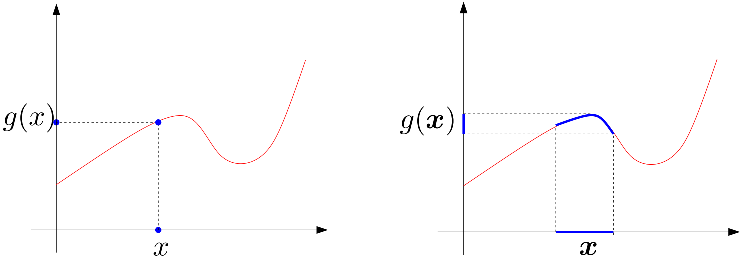
UPPSALA UNIVERSITET

Set-valued computations

Interval analysis

All our computations are set-valued, and are based on the *inclusion principle*:

$$\mathrm{range}(g; \boldsymbol{x}) = \{g(x) \colon x \in \boldsymbol{x}\} \subseteq g(\boldsymbol{x})$$



Interval Computations Web Page

`http://www.cs.utep.edu/interval-comp`

Set-valued computations

Points versus sets in parameter space

We move from the *point-valued* model function $f(x; p)$ to the *set-valued* version $f(x; \boldsymbol{p})$.

Set-valued computations

Points versus sets in parameter space

We move from the *point-valued* model function $f(x;p)$ to the *set-valued* version $f(x;\boldsymbol{p})$.

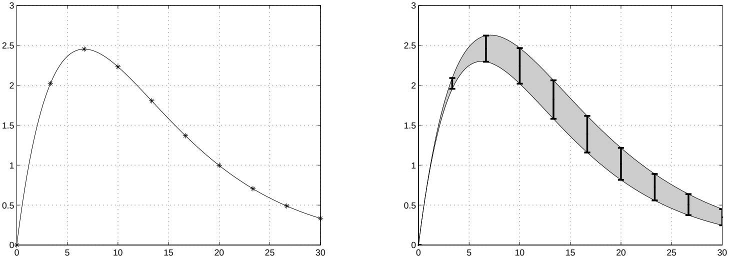


Figure: (a) $p = 0.15$, a point in $\mathcal{P}$. (b) $\boldsymbol{p} = [0.14, 0.16]$, a subset of $\mathcal{P}$. The model function is $f(x;p) = xe^{-px}$, and 10 samples are shown.

Parameter reconstruction

Strategy

Adaptively bisect the parameter space into sub-boxes: $\mathcal{P} = \cup_{j=1}^{K} \boldsymbol{p}_j$ and examine each $\boldsymbol{p}_j$ separately.

UPPSALA UNIVERSITET

Parameter reconstruction

Strategy

Adaptively bisect the parameter space into sub-boxes: $\mathcal{P} = \cup_{j=1}^{K} \boldsymbol{p}_j$ and examine each $\boldsymbol{p}_j$ separately.

We will associate each sub-box $\boldsymbol{p}$ of the parameter space to one of three categories:

Parameter reconstruction

Strategy

Adaptively bisect the parameter space into sub-boxes: $\mathcal{P} = \cup_{j=1}^{K} \boldsymbol{p}_j$ and examine each $\boldsymbol{p}_j$ separately.

We will associate each sub-box $\boldsymbol{p}$ of the parameter space to one of three categories:

(1) consistent

if $f(x_i; \boldsymbol{p}) \subset \boldsymbol{y}_i$ for all $i = 0, \ldots, N$. SAVE

Parameter reconstruction

Strategy

Adaptively bisect the parameter space into sub-boxes: $\mathcal{P} = \cup_{j=1}^{K} \boldsymbol{p}_j$ and examine each $\boldsymbol{p}_j$ separately.

We will associate each sub-box $\boldsymbol{p}$ of the parameter space to one of three categories:

(1) consistent

if $f(x_i; \boldsymbol{p}) \subset \boldsymbol{y}_i$ for all $i = 0, \dots, N$. SAVE

(2) inconsistent

if $f(x_i; \boldsymbol{p}) \cap \boldsymbol{y}_i = \emptyset$ for at least one i. TRASH

UPPSALA UNIVERSITET

Parameter reconstruction

Strategy

Adaptively bisect the parameter space into sub-boxes: $\mathcal{P} = \cup_{j=1}^{K} \boldsymbol{p}_j$ and examine each $\boldsymbol{p}_j$ separately.

We will associate each sub-box $\boldsymbol{p}$ of the parameter space to one of three categories:

(1) consistent

if $f(x_i; \boldsymbol{p}) \subset \boldsymbol{y}_i$ for all $i = 0, \ldots, N$. SAVE

(2) inconsistent

if $f(x_i; \boldsymbol{p}) \cap \boldsymbol{y}_i = \emptyset$ for at least one i. TRASH

(3) undetermined

not (1), but $f(x_i; \boldsymbol{p}) \cap \boldsymbol{y}_i \neq \emptyset$ for all $i = 0, \ldots, N$. SPLIT

Parameter reconstruction

Example

Consider the model function

$$f(x; p_1, p_2) = 5e^{-p_1 x} - 4 \times 10^{-6} e^{-p_2 x}$$

with samples taken at $x = 0, 5 \ldots, 40$ using $p^\star = (0.11, -0.32)$. With a relative noise level of 90%, we get the following set of consistent parameters:

UPPSALA UNIVERSITET

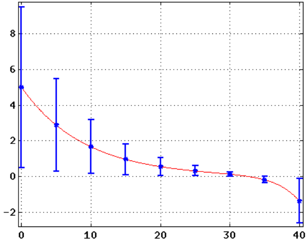
Parameter reconstruction

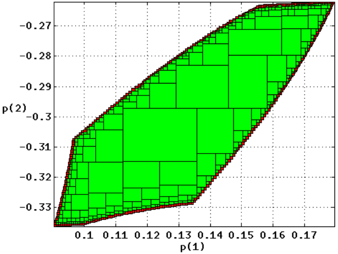
Example

Consider the model function

$$f(x; p_1, p_2) = 5e^{-p_1 x} - 4 \times 10^{-6} e^{-p_2 x}$$

with samples taken at $x = 0, 5 \ldots, 40$ using $p^\star = (0.11, -0.32)$. With a relative noise level of 90%, we get the following set of consistent parameters:

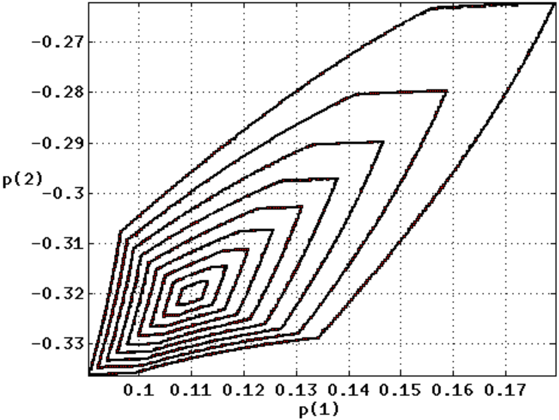




UPPSALA UNIVERSITET

Parameter reconstruction

Varying the relative noise levels between $10, 20\ldots, 90\%$, we get the following indeterminate sets.



Constraint propagation

Constraining the parameter/data space

UPPSALA
UNIVERSITET

Constraint propagation

Constraining the parameter/data space

- **Q:** How do we speed up the estimation process?

Constraint propagation

Constraining the parameter/data space

- **Q:** How do we speed up the estimation process?
- **A:** By quickly discarding inconsistent parameters/data!

UPPSALA
UNIVERSITET

Constraint propagation

Constraining the parameter/data space

- **Q:** How do we speed up the estimation process?
- **A:** By quickly discarding inconsistent parameters/data!
- **Q:** How do we do that?

Constraint propagation

Constraining the parameter/data space

- **Q:** How do we speed up the estimation process?
- **A:** By quickly discarding inconsistent parameters/data!
- **Q:** How do we do that?
- **A:** By set-valued constraint propagation!

Constraint propagation

Constraining the parameter/data space

- **Q:** How do we speed up the estimation process?
- **A:** By quickly discarding inconsistent parameters/data!
- **Q:** How do we do that?
- **A:** By set-valued constraint propagation!

Example

Let $f(x;p) = xe^{-px}$, and consider the situation $\boldsymbol{p} = [0, 1]$ and $(x, \boldsymbol{y}) = (2, [1, 3])$.

Constraint propagation

Constraining the parameter/data space

- **Q:** How do we speed up the estimation process?
- **A:** By quickly discarding inconsistent parameters/data!
- **Q:** How do we do that?
- **A:** By set-valued constraint propagation!

Example

Let $f(x;p) = xe^{-px}$, and consider the situation $\boldsymbol{p} = [0,1]$ and $(x, \boldsymbol{y}) = (2, [1,3])$. By a forward evaluation, we have

$$f(2;[0,1]) = 2e^{-2[0,1]} = 2e^{[-2,0]} = 2[e^{-2}, 1] = [2e^{-2}, 2].$$

Constraint propagation

Constraining the parameter/data space

- **Q:** How do we speed up the estimation process?
- **A:** By quickly discarding inconsistent parameters/data!
- **Q:** How do we do that?
- **A:** By set-valued constraint propagation!

Example

Let $f(x;p) = xe^{-px}$, and consider the situation $\boldsymbol{p} = [0,1]$ and $(x, \boldsymbol{y}) = (2, [1,3])$. By a forward evaluation, we have

$$f(2;[0,1]) = 2e^{-2[0,1]} = 2e^{[-2,0]} = 2[e^{-2},1] = [2e^{-2},2].$$

This allows us to contract the data range according to

$$\boldsymbol{y} \mapsto \boldsymbol{y} \cap f(x;\boldsymbol{p}) = [1,3] \cap [2e^{-2},2] = [1,2].$$

Constraint propagation

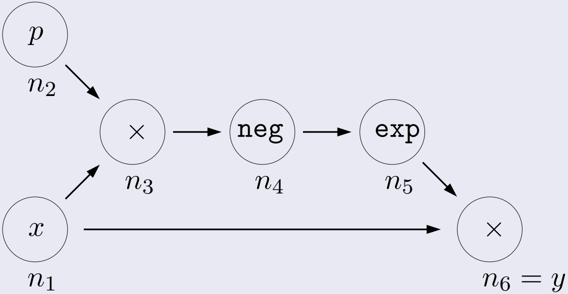
Directed Acyclic Graphs (DAGs)

We use a DAG representation of the model function to automate constraint propagations.

Constraint propagation

Directed Acyclic Graphs (DAGs)

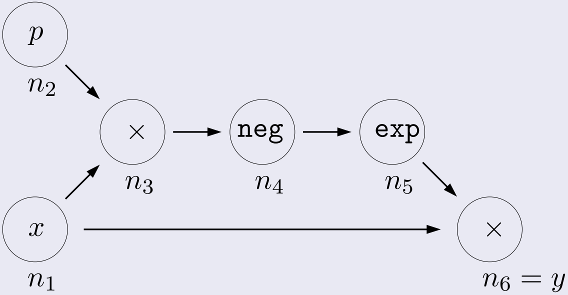
We use a DAG representation of the model function to automate constraint propagations.



Constraint propagation

Directed Acyclic Graphs (DAGs)

We use a DAG representation of the model function to automate constraint propagation.



$$
\begin{aligned}
n_1 &= x \\
n_2 &= p \\
n_3 &= n_1 \times n_2 \\
n_4 &= -n_3 \\
n_5 &= e^{n_4} \\
n_6 &= n_1 \times n_5.
\end{aligned}
$$

Figure: The DAG representation of a forward sweep of $y = xe^{-px}$, together with the corresponding code list.

Constraint propagation

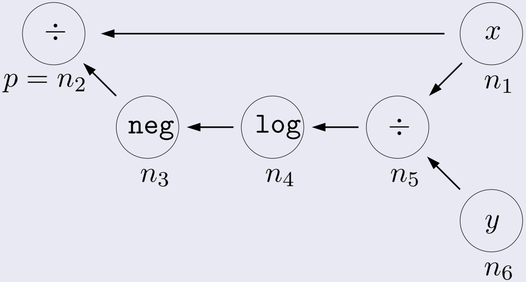
Directed Acyclic Graphs (DAGs)

We can propagate constraints from data to the parameter by moving *backwards* in the code list.

Constraint propagation

Directed Acyclic Graphs (DAGs)

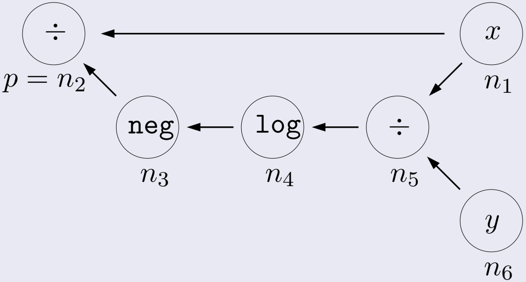
We can propagate constraints from data to the parameter by moving *backwards* in the code list.



Constraint propagation

Directed Acyclic Graphs (DAGs)

We can propagate constraints from data to the parameter by moving *backwards* in the code list.



$$n_5 = n_6 \div n_1$$
$$n_4 = \log n_5$$
$$n_3 = -n_4$$
$$n_2 = n_3 \div n_1.$$

Figure: The DAG representation of a backward sweep of $y = xe^{-px}$, together with the corresponding code list.

Constraint propagation

Example

Again, we work on the model function $y = f(x;p) = xe^{-px}$, but now with the data $(x, \boldsymbol{y}) = (2, [1, 3])$, together with the parameter domain $\boldsymbol{p} = [0, 1]$.

Constraint propagation

Example

Again, we work on the model function $y = f(x;p) = xe^{-px}$, but now with the data $(x, \boldsymbol{y}) = (2, [1,3])$, together with the parameter domain $\boldsymbol{p} = [0,1]$. The forward sweep, performed in Example 2, contracts the interval data to $\boldsymbol{y} = [1,2]$.

UPPSALA UNIVERSITET

Constraint propagation

Example

Again, we work on the model function $y = f(x; p) = xe^{-px}$, but now with the data $(x, \boldsymbol{y}) = (2, [1, 3])$, together with the parameter domain $\boldsymbol{p} = [0, 1]$. The forward sweep, performed in Example 2, contracts the interval data to $\boldsymbol{y} = [1, 2]$. Performing a backward sweep contracts the interval parameter to $\boldsymbol{p} = [0, \frac{1}{2} \log 2]$:

$$\begin{array}{lllllll} n_5 & = & n_6 \div n_1 & = & [1, 2] \div 2 & = & [\frac{1}{2}, 1] \\ n_4 & = & \log n_5 & = & \log [\frac{1}{2}, 1] & = & [-\log 2, 0] \\ n_3 & = & -n_4 & = & [0, \log 2] & & \\ n_2 & = & n_3 \div n_1 & = & \frac{1}{2}[0, \log 2] & \approx & [0, 0.34657359]. \end{array}$$

UPPSALA UNIVERSITET

Constraint propagation

Example

Again, we work on the model function $y = f(x;p) = xe^{-px}$, but now with the data $(x, \boldsymbol{y}) = (2, [1,3])$, together with the parameter domain $\boldsymbol{p} = [0,1]$. The forward sweep, performed in Example 2, contracts the interval data to $\boldsymbol{y} = [1,2]$. Performing a backward sweep contracts the interval parameter to $\boldsymbol{p} = [0, \frac{1}{2}\log 2]$:

$$\begin{array}{lllllll} n_5 & = & n_6 \div n_1 & = & [1,2] \div 2 & = & [\frac{1}{2}, 1] \\ n_4 & = & \log n_5 & = & \log [\frac{1}{2}, 1] & = & [-\log 2, 0] \\ n_3 & = & -n_4 & = & [0, \log 2] & & \\ n_2 & = & n_3 \div n_1 & = & \frac{1}{2}[0, \log 2] & \approx & [0, 0.34657359]. \end{array}$$

Note that, in one forward/backward sweep, we managed to exclude over 65% of the parameter domain, at the same time reducing the data uncertainty by 50%.

Mixed-effects models

Mixed-effects models

We are given several data sets (trajectories) corresponding to k different "individuals":

$$
\begin{aligned}
\text{individual}_1 : \quad & (x_{11}, y_{11}), (x_{12}, y_{12}), \ldots, (x_{1N}, y_{1N_1}) \\
\text{individual}_2 : \quad & (x_{21}, y_{21}), (x_{22}, y_{22}), \ldots, (x_{2N}, y_{2N_2}) \\
& \quad \vdots \qquad\qquad \vdots \qquad\qquad\quad \vdots \\
\text{individual}_k : \quad & (x_{k1}, y_{k1}), (x_{k2}, y_{k2}), \ldots, (x_{kN}, y_{kN_k}).
\end{aligned}
$$

Some model parameters are equal (shared) for all individuals, and some are distinct.

UPPSALA UNIVERSITET

Mixed-effects models

Mixed-effects models

We are given several data sets (trajectories) corresponding to k different "individuals":

$$\begin{array}{ll} \text{individual}_1: & (x_{11}, y_{11}), (x_{12}, y_{12}), \ldots, (x_{1N}, y_{1N_1}) \\ \text{individual}_2: & (x_{21}, y_{21}), (x_{22}, y_{22}), \ldots, (x_{2N}, y_{2N_2}) \\ & \quad\vdots \qquad\qquad \vdots \qquad\qquad\qquad \vdots \\ \text{individual}_k: & (x_{k1}, y_{k1}), (x_{k2}, y_{k2}), \ldots, (x_{kN}, y_{kN_k}). \end{array}$$

Some model parameters are equal (shared) for all individuals, and some are distinct.

- We need to consider all individuals simultaneously. Otherwise the number of unknown parameters may be too large.

A mixed-effects model for orange tree truncs

Example

We will apply our method to the following scenario:

A mixed-effects model for orange tree truncs

Example

We will apply our method to the following scenario:

Model function: $$f(x;p) = \frac{p_1}{1 + p_2 e^{p_3 x}}$$

A mixed-effects model for orange tree truncs

Example

We will apply our method to the following scenario:

Model function: $$f(x;p) = \frac{p_1}{1 + p_2 e^{p_3 x}}$$

Individual parameter: $$p_{i1} = p_1^{\sharp} + \eta_i, \quad \eta_i \sim N(0, \sigma^2)$$

UPPSALA UNIVERSITET

A mixed-effects model for orange tree truncs

Example

We will apply our method to the following scenario:

Model function:	$f(x;p) = \dfrac{p_1}{1 + p_2 e^{p_3 x}}$
Individual parameter:	$p_{i1} = p_1^\sharp + \eta_i, \quad \eta_i \sim N(0, \sigma^2)$
Data perturbation:	$y_{ij} = y_{ij}^\sharp(1 + \theta_{ij}), \quad \theta_{ij} \sim U(-\epsilon, +\epsilon)$

A mixed-effects model for orange tree truncs

Example

We will apply our method to the following scenario:

Model function:	$f(x;p) = \dfrac{p_1}{1 + p_2 e^{p_3 x}}$
Individual parameter:	$p_{i1} = p_1^\sharp + \eta_i, \quad \eta_i \sim N(0, \sigma^2)$
Data perturbation:	$y_{ij} = y_{ij}^\sharp (1 + \theta_{ij}), \quad \theta_{ij} \sim U(-\epsilon, +\epsilon)$

For this specific example, we will use $N_p \in \{1, 2, 5, 50\}$ subjects, sampled at $N_d = 10$ data sites, evenly spaced within $[100, 1600]$.

A mixed-effects model for orange tree truncs

Example

We will apply our method to the following scenario:

$$\text{Model function:} \qquad f(x;p) = \frac{p_1}{1 + p_2 e^{p_3 x}}$$

$$\text{Individual parameter:} \qquad p_{i1} = p_1^\sharp + \eta_i, \quad \eta_i \sim N(0, \sigma^2)$$

$$\text{Data perturbation:} \qquad y_{ij} = y_{ij}^\sharp (1 + \theta_{ij}), \quad \theta_{ij} \sim U(-\epsilon, +\epsilon)$$

For this specific example, we will use $N_p \in \{1, 2, 5, 50\}$ subjects, sampled at $N_d = 10$ data sites, evenly spaced within $[100, 1600]$.

Target parameters:

$p^\sharp = (191.84, 8.153, -0.0029)$, $\sigma = 20$, $\epsilon \in \{0.01, 0.1, 0.2, 0.5\}$.

UPPSALA UNIVERSITET

A mixed-effects model for orange tree truncs

Example

We will apply our method to the following scenario:

$$\text{Model function:} \quad f(x;p) = \frac{p_1}{1 + p_2 e^{p_3 x}}$$

$$\text{Individual parameter:} \quad p_{i1} = p_1^\sharp + \eta_i, \quad \eta_i \sim N(0, \sigma^2)$$

$$\text{Data perturbation:} \quad y_{ij} = y_{ij}^\sharp (1 + \theta_{ij}), \quad \theta_{ij} \sim U(-\epsilon, +\epsilon)$$

For this specific example, we will use $N_p \in \{1, 2, 5, 50\}$ subjects, sampled at $N_d = 10$ data sites, evenly spaced within $[100, 1600]$.

Target parameters:

$p^\sharp = (191.84, 8.153, -0.0029)$, $\sigma = 20$, $\epsilon \in \{0.01, 0.1, 0.2, 0.5\}$.

Search region:

$$\mathcal{P} = ([0, 300], [0, 9], [-1, 0]).$$

UPPSALA UNIVERSITET

A mixed-effects model for orange tree truncs

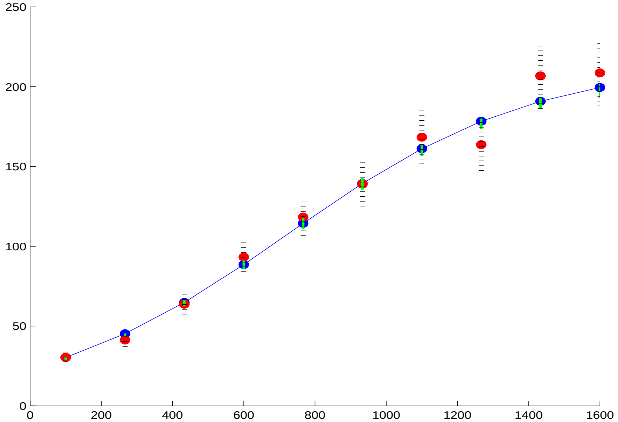


Figure: Data inflation and contraction for the example. The graph of the model function for one subject (blue line). The data points are marked with red dots. The inflated data sets are shown as striped bars, and the re-contracted data as green bars.

UPPSALA UNIVERSITET

A mixed-effects model for orange tree truncs

Numerical results

	$N_p = 1$	$N_p = 2$
$\epsilon = 0.01$	190.639 (– –) (0.010)	193.141 (19.6) (0.013)
$\epsilon = 0.1$	194.139 (– –) (0.092)	195.233 (21.1) (0.097)
$\epsilon = 0.2$	189.139 (– –) (0.190)	193.437 (20.3) (0.192)
$\epsilon = 0.5$	167.226 (– –) (0.604)	167.770 (26.6) (0.589)

	$N_p = 5$	$N_p = 50$
$\epsilon = 0.01$	191.675 (20.1) (0.014)	191.239 (20.1) (0.012)
$\epsilon = 0.1$	192.954 (21.4) (0.099)	198.428 (22.2) (0.110)
$\epsilon = 0.2$	191.773 (20.3) (0.203)	197.580 (23.6) (0.214)
$\epsilon = 0.5$	164.656 (23.9) (0.620)	174.318 (27.1) (0.618)

Table: The results of four experiments for the example, each using 100 trial runs with $p_1 = 191.184$, and $\sigma = 20.0$. For each pair (ϵ, N_p), we display the triple $\mu(p_1)$, $\mu(\sigma)$, and $\mu(\epsilon)$ – the average estimates of the distribution parameters for p_1, and the data error.

A mixed-effects model for orange tree truncs

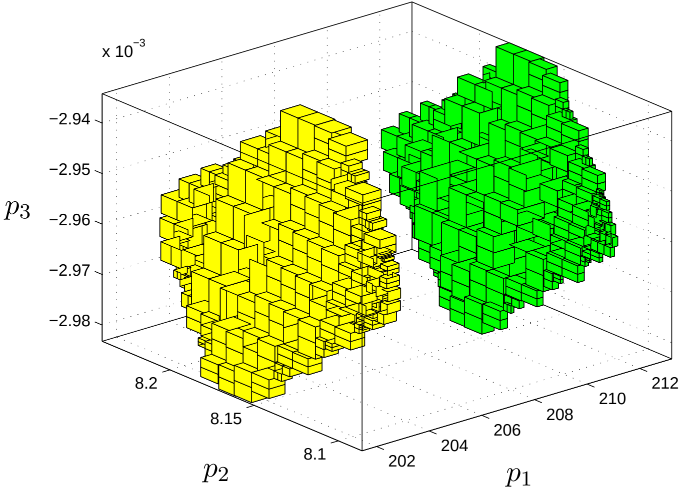


Figure: The set of consistent parameters for two subjects from the example.

UPPSALA UNIVERSITET

Summary

Main steps

Summary

Main steps

- Relax the problem via data inflation,

UPPSALA
UNIVERSITET

Summary

Main steps

- Relax the problem via data inflation,
- Reduce the data and parameter sets via constraint propagation,

UPPSALA
UNIVERSITET

Summary

Main steps

- Relax the problem via data inflation,
- Reduce the data and parameter sets via constraint propagation,
- Produce traditional statistics via data gridding/sampling.

Summary

Main steps

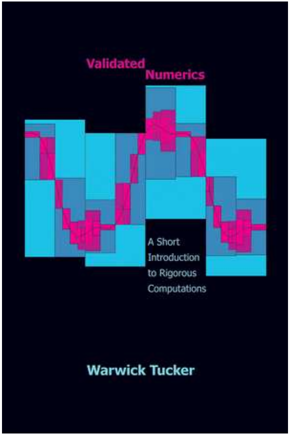
- Relax the problem via data inflation,
- Reduce the data and parameter sets via constraint propagation,
- Produce traditional statistics via data gridding/sampling.

CAPA

Computer-Aided Proofs in Analysis Web Page

http://www2.math.uu.se/~warwick/CAPA/

A short message from your sponsors...



Validated Numerics:
A Short Introduction to Rigorous Computations

Warwick Tucker

Princeton University Press, 2011

ISBN: 9780691147819

152 pp.|6 x 9|41 illus.|12 tables.

USD 45.00/GBP 30.95

`http://press.princeton.edu/titles/9488.html`