Parameter Estimation of Mathematical Models Described by Differential Equations

Hossein Zivaripiran

Department of Computer Science
University of Toronto
Outline

- Modeling with Differential Equations (IVPs, DDEs)
- Modeling and Parameter Estimation
- Numerical Parameter Estimation of IVPs
- DDEs and Parameter Estimation
An Initial Value Problem (IVP) for Ordinary Differential Equations (ODEs)

\[
\begin{align*}
    y'(t) &= f(t, y(t)) \\
    y(t_0) &= y_0
\end{align*}
\]
Modeling with DE - Formulation

- An Initial Value Problem (IVP) for Ordinary Differential Equations (ODEs)

\[ y'(t) = f(t, y(t)) \]
\[ y(t_0) = y_0 \]

- Retarded Delay Differential Equations (RDDEs)

\[ y'(t) = f(t, y(t), y(t - \sigma_1), \ldots, y(t - \sigma_\nu)) \text{ for } t_0 \leq t \leq t_F \]
\[ y(t) = \phi(t), \text{ for } t \leq t_0 \]

\( \sigma_i = \sigma_i(t, y(t)) \geq 0 \) delay (constant / time dependent / state dependent)
\( \phi(t) \) history function (constant / time dependent)
An Initial Value Problem (IVP) for Ordinary Differential Equations (ODEs)

\[ y'(t) = f(t, y(t)) \]
\[ y(t_0) = y_0 \]

Retarded Delay Differential Equations (RDDEs)

\[ y'(t) = f(t, y(t), y(t - \sigma_1), \ldots, y(t - \sigma_{\nu})) \quad \text{for } t_0 \leq t \leq t_F \]
\[ y(t) = \phi(t), \quad \text{for } t \leq t_0 \]

\[ \sigma_i = \sigma_i(t, y(t)) \geq 0 \quad \text{delay (constant / time dependent / state dependent)} \]
\[ \phi(t) \quad \text{history function (constant / time dependent)} \]

Neutral Delay Differential Equations (NDDEs)

\[ y'(t) = f(t, y(t), y(t - \sigma_1), \ldots, y(t - \sigma_{\nu}), \quad y'(t - \sigma_{\nu+1}), \ldots, y'(t - \sigma_{\nu+\omega})) \quad \text{for } t_0 \leq t \leq t_F \]
\[ y(t) = \phi(t), \quad y'(t) = \phi'(t), \quad \text{for } t \leq t_0, \]
The Van der Pol Oscillator,

\[
\frac{d^2x}{dt^2} - \mu(1 - x^2) \frac{dx}{dt} + x = 0
\]

using \( y_1 = x \) and \( y_2 = \frac{dx}{dt} \),

\[
\begin{align*}
y_1'(t) &= y_2(t) \\
y_2'(t) &= \mu(1 - y_1^2)y_2 - y_1
\end{align*}
\]
The Van der Pol Oscillator with $\mu = 1, y_1(0) = 2, y_2(0) = 0$
A neutral delay logistic Gause-type predator-prey system [Kuang 1991]

\[ y_1'(t) = y_1(t)(1 - y_1(t - \tau) - \rho y_1'(t - \tau)) - \frac{y_2(t)y_1(t)^2}{y_1(t)^2 + 1} \]

\[ y_2'(t) = y_2(t) \left( \frac{y_1(t)^2}{y_1(t)^2 + 1} - \alpha \right) \]

where \( \alpha = 1/10 \), \( \rho = 29/10 \) and \( \tau = 21/50 \), for \( t \) in \([0, 30]\). The history functions are

\[ \phi_1(t) = \frac{33}{100} - \frac{1}{10}t \]

\[ \phi_2(t) = \frac{111}{50} + \frac{1}{10}t \]

for \( t \leq 0 \).
The predator-prey model
Parameterized Models

♦ A parameterized IVP

\[ y'(t; p) = f(t, y(t; p); p) \]
\[ y(t_0) = y_0(p) \]

For example, \( p = [\mu] \) in the Van der Pol oscillator

\[ y_1'(t) = y_2(t) \]
\[ y_2'(t) = \mu(1 - y_1^2)y_2 - y_1 \]

♦ A simple parameterized DDE

\[ y'(t; p) = f(t, y(t; p), y(t - \sigma(t; p)); p) \text{ for } t_0(p) \leq t \]
\[ y(t; p) = \phi(t; p), \text{ for } t \leq t_0(p) \]
Parameter Estimation Problem

♦ A System of Parameterized IVP

\[ y'(t; p) = f(t, y(t; p); p) \]
\[ y(t_0) = y_0(p) \]

or DDE

\[ y'(t; p) = f(t, y(t; p), y(t - \sigma(t; p)); p) \quad \text{for} \quad t_0(p) \leq t \]
\[ y(t; p) = \phi(t; p), \quad \text{for} \quad t \leq t_0(p) \]

♦ A Set of Data (Observations/Measurements)

\[ \{Y(\gamma_i) \approx y(\gamma_i; p^*)\} \]

♦ Estimate \( p^* \) by minimizing an objective function.

E.g.

\[ W(p) = \sum_{i} [Y(\gamma_i) - y(\gamma_i; p)]^2. \]
Modeling and Parameter Estimation - Importance

Physical/Biological Phenomenon

Physical Laws / Empirical Rules

Mathematical Model

Sensitivity Analysis

Refined Mathematical Model

Parameter Estimation

Practical Mathematical Model
Algorithms for Nonlinear Least-Squares

♦ Unconstrained

\[
\min_{p} W(p) = \sum_{i} [Y(\gamma_i) - y(\gamma_i; p)]^2.
\]

\[\downarrow\]

Levenberg-Marquardt

Variations of Sequential Quadratic Programming (SQP)

♦ Constrained

\[
\min_{p} W(p) = \sum_{i} [Y(\gamma_i) - y(\gamma_i; p)]^2,
\]

\[
c_j(p) = 0, \quad j \in \mathcal{E},
\]

\[
c_j(p) \geq 0, \quad j \in \mathcal{I}.
\]

\[\downarrow\]

Sequential Quadratic Programming (SQP)
The initial value approach:

1. Choose an initial guess for the parameters
2. Solve model equations
3. Check optimality conditions, (if satisfied ⇒ stop).
4. Choose a better value for the parameters and continue with (2)

The main difficulty: If the parameters are far from the correct ones the trial trajectory soon loses contact to the measurements.

If the optimization method needs to compute the gradient or the Hessian of the objective function,

\[
\left( \frac{\partial W(p)}{\partial p_l} \right) = -2 \sum_i \left[ Y(\gamma_i) - y(\gamma_i; p) \right] \left( \frac{\partial y(\gamma_i; p)}{\partial p_l} \right)
\]

\[
\left( \frac{\partial^2 W(p)}{\partial p_l \partial p_m} \right) = 2 \sum_i \left[ \left( \frac{\partial y(\gamma_i; p)}{\partial p_l} \right) \left( \frac{\partial y(\gamma_i; p)}{\partial p_m} \right) - \left[ Y(\gamma_i) - y(\gamma_i; p) \right] \left( \frac{\partial^2 y(\gamma_i; p)}{\partial p_l \partial p_m} \right) \right]
\]

the sensitivity equations are usually used to provide the required values. An alternative approach is to use a divided-difference approximation.
A Simple Example,

- The IVP

\[
y_1'(t) = ay_2(t) \\
y_2'(t) = -ay_1(t)
\]

with initial conditions

\[
y_1(0) = 0 \\
y_2(0) = 1
\]

- the analytical solution is

\[
y_1(t) = \sin(at) \\
y_2(t) = \cos(at)
\]
Numerical Parameter Estimation of IVPs

\[ y_1(t) = a^* \]

- \( a = 1 \)
- \( a = 1.05 \)
- \( a = 1.1 \)
- \( a = 1.15 \)
- \( a = 1.45 \)
- \( a = 1.5 \)
- \( a = 1.55 \)
Multiple shooting:

- Partitions the fitting interval into many subintervals, each having its own initial values.
- The measurements are used to get starting guesses for initial values of each interval.
- In an iterative process, the algorithm minimizes $W(p)$ on the one hand and enforces the continuity of the full trajectory on the other hand.

Advantages: Divergences are avoided and the danger of local minima is reduced.
The adapted initial value approach [Paul, 1997]:
Assume that jumps in the derivative of $y(t; p)$ with respect to $t$ occur at the points

$$\Lambda(p) \equiv \{\lambda_1(p), \lambda_2(p), \ldots\}.$$ 

Such discontinuities, when arising from the initial point $t_0(p)$ (and the initial function $\phi(t; p)$), may propagate into $W(p)$ via the solution values $\{y(\gamma_i; p)\}$.

The first and second order partial derivatives of the objective function are

$$\left(\frac{\partial W(p)}{\partial p_l}\right)_\pm = -2 \sum_i [Y(\gamma_i) - y(\gamma_i; p)] \left(\frac{\partial y(\gamma_i; p)}{\partial p_l}\right)_\pm$$

$$\left(\frac{\partial^2 W(p)}{\partial p_l \partial p_m}\right)_\pm = 2 \sum_i \left[ \left(\frac{\partial y(\gamma_i; p)}{\partial p_l}\right)_\pm \left(\frac{\partial y(\gamma_i; p)}{\partial p_m}\right)_\pm - [Y(\gamma_i) - y(\gamma_i; p)] \left(\frac{\partial^2 y(\gamma_i; p)}{\partial p_l \partial p_m}\right)_\pm \right]$$

A general rule for the propagation of discontinuities to $W(p)$ [Baker & Paul, 1997]: if a discontinuity point $\lambda_r(p)$ coincides with one of the data points $\gamma_i$, and $\lambda_r(p)$ varies as some parameters vary, then $W(p)$ has a jump in its partial derivatives that correspond to the varying parameters.
**Multiple shooting** [Horbet et al., 2002]:

Use cubic splines to parameterize the initial curves and formulate the continuity of the trajectory in terms of these spline variables.

A two-phase procedure:

- During the first iterations of the optimization, the spline variables are held fixed because they are expected to be estimated well from the data.
- After the algorithm has converged for the first time, they are released and fitted together with the other variables until the final convergence is achieved.

Can attain higher convergence rate in the case of noisy data.
The full discretization approach [Murphy, 1990]:

Use linear splines to describe solution and delay functions.

As the result the problem becomes a very large minimization problem.

The number of mesh points is increased gradually to be able to satisfy the specified error tolerance.

Although the method has the generality of dealing with any kinds of unknown parameters, it suffers from the heavy computations, slow convergence rate and possibility of being trapped in a local minimum.
Non-smooth optimization

Consider the continuous function

\[ y(t) = \begin{cases} 
-5(t - \tau) + c, & \text{if } t < \tau \\
5(t - \tau) + c, & \text{if } t \geq \tau 
\end{cases} \]

with discontinuous derivative

\[ y'(t) = \begin{cases} 
-5, & \text{if } t < \tau \\
5, & \text{if } t \geq \tau 
\end{cases} \]

and observed value of \( y \) at the discontinuity point \( \tau^* = 4 \).
DDEs and Parameter Estimation

\[ W(\tau) \]
\[ \frac{\partial W(\tau)}{\partial \tau} \]
- Try to find $\tau^*$ using MATLAB’s unconstrained minimization routine $\textit{fminunc}$

\[
\downarrow
\]

31 iterations.

- Try to use MATLAB’s constrained minimization routine $\textit{fmincon}$ with the added constraint $\tau \leq 4$

\[
\downarrow
\]

2 iterations.

- Try to use MATLAB’s constrained minimization routine $\textit{fmincon}$ with the added constraint $\tau \geq 4$

\[
\downarrow
\]

2 iterations.
Appearance of Non-smoothness

Partial derivatives (gradient) of the objective function

\[
\left( \frac{\partial W(p)}{\partial p_i} \right)_\pm = -2 \sum_i \left[ Y(\gamma_i) - y(\gamma_i; p) \right] \left( \frac{\partial y(\gamma_i; p)}{\partial p_l} \right)_\pm
\]

\[
\left( \frac{\partial^2 W(p)}{\partial p_i \partial p_m} \right)_\pm = 2 \sum_i \left[ \left( \frac{\partial y(\gamma_i; p)}{\partial p_l} \right)_\pm \left( \frac{\partial y(\gamma_i; p)}{\partial p_m} \right)_\pm - \left[ Y(\gamma_i) - y(\gamma_i; p) \right] \left( \frac{\partial^2 y(\gamma_i; p)}{\partial p_l \partial p_m} \right)_\pm \right]
\]

Recall the jump equation for sensitivities

\[
\frac{\partial y}{\partial p_l}(\lambda^+_{r+1}) = \frac{\partial y}{\partial p_l}(\lambda^-_{r+1}) + \left( y'(\lambda^-_{r+1}) - y'(\lambda^+_{r+1}) \right) \frac{\partial \lambda_{r+1}(p)}{\partial p_l}
\]

The General Rule: A jump occurs in \( W(p) \) when

a discontinuity point \( \lambda_{r+1}(p) \) passes a data point \( \gamma_i \)
Algorithms for Nonlinear Least-Squares

- Unconstrained

\[
\min_{\mathbf{p}} W(\mathbf{p}) = \sum_{i} [Y(\gamma_i) - y(\gamma_i; \mathbf{p})]^2.
\]

\[
\downarrow
\]

Levenberg-Marquardt

Variations of Sequential Quadratic Programming (SQP)

- Constrained

\[
\min_{\mathbf{p}} W(\mathbf{p}) = \sum_{i} [Y(\gamma_i) - y(\gamma_i; \mathbf{p})]^2,
\]

\[
c_j(\mathbf{p}) = 0, \quad j \in \mathcal{E},
\]

\[
c_j(\mathbf{p}) \geq 0, \quad j \in \mathcal{I}.
\]

\[
\downarrow
\]

Sequential Quadratic Programming (SQP)

The smoothness of functions involved in the problem, the objective function and constraints, is a necessary assumption.
Avoiding The Non-smoothness

Force the ordering by adding $\lambda_r(p) \leq \gamma_i \leq \lambda_{r+1}(p)$ to the set of constraints.

The partial derivatives (gradient) of the new constraints, $\frac{\partial \lambda_{r+1}(p)}{\partial p}$, can be computed recursively.
Safe Conduction of Optimization

Let

\[ ContinuityConstarints[p_{Base}] = \{ \lambda_{r_{p_{Base}}} (p) \leq \gamma_i \leq \lambda_{r_{p_{Base}}+1}(p) \} \]

Then we can describe the steps for finding a local optimum as the following

1. Start with \( p_c \leftarrow p_0 \).
2. \( p_{new} \leftarrow SQP(p_c, ContinuityConstarints[p_c]) \)
3. If some of \( ContinuityConstarints[p_c] \) are active, then \( p_c \leftarrow p_{new} \) and continue at (2), otherwise stop with \( p^* = p_{new} \).
Estimating $\tau$ for the predator-prey model

\[
\begin{align*}
y'_1(t) &= y_1(t)(1 - y_1(t - \tau) - \rho y'_1(t - \tau)) - \frac{y_2(t)y_1(t)^2}{y_1(t)^2 + 1} \\
y'_2(t) &= y_2(t)\left(\frac{y_1(t)^2}{y_1(t)^2 + 1} - \alpha\right)
\end{align*}
\]

Start with up to 10% random perturbation in original $\tau$, and up to 3% randomly perturbed $y(t; \tau)$ as Data ($Y$).

For $\gamma$'s we choose 10 random points, one of which is a discontinuity point.

Run the parameter estimator 3 times.

Results

<table>
<thead>
<tr>
<th>Estimator Choice</th>
<th>FCN</th>
<th>OBJ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very Simple</td>
<td>72,2697</td>
<td>0.000142917</td>
</tr>
<tr>
<td>Using Sensitivities</td>
<td>26,942</td>
<td>0.000142917</td>
</tr>
<tr>
<td>Adding Constraints</td>
<td>9,916</td>
<td>0.000142917</td>
</tr>
</tbody>
</table>
Back to First Talk!