Inverse problems in biological systems

Parameter identification in Biochemical Pathways

Nathaniel Mays

Department of Mathematics
University of Pittsburgh

February 20, 2009

Outline

Introduction

An ill-posed problem is one that

- has no solution,
- if it has a solution, it isn't unique, or
- the solution does not depend continuously on the data.

Introduction

An ill-posed problem is one that

- has no solution,
- if it has a solution, it isn't unique, or
- the solution does not depend continuously on the data.

A forward (or classical) problem is one where given model parameters, you can find a solution.

An inverse problem is one where you have the data, but not the parameters that gave the data.

Introduction

An ill-posed problem is one that

- has no solution,
- if it has a solution, it isn't unique, or
- the solution does not depend continuously on the data.

A forward (or classical) problem is one where given model parameters, you can find a solution.

An inverse problem is one where you have the data, but not the parameters that gave the data.

Examples of ill posed inverse problems would be:

- Medical Imaging
- Stream Pollution
- Parameter Identification

Parameter Identification

You're given the autonomous initial value problem

$$y'(t) = f(y(t), x, \tilde{x})$$
$$y(0) = y_0.$$

You have observed measurements y and some known parameters $\tilde{x}$
Some questions to ask are:

Parameter Identification

You're given the autonomous initial value problem

$$y'(t) = f(y(t), x, \tilde{x})$$
$$y(0) = y_0.$$

You have observed measurements y and some known parameters $\tilde{x}$

Some questions to ask are:

- Can you find what parameters x are needed to obtain that data?

Parameter Identification

You're given the autonomous initial value problem

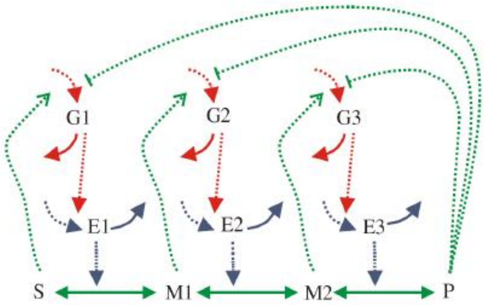
$$y'(t) = f(y(t), x, \tilde{x})$$
$$y(0) = y_0.$$

You have observed measurements y and some known parameters $\tilde{x}$
Some questions to ask are:

- Can you find what parameters x are needed to obtain that data?
- Now what if those measurements have noise?

Benchmark Problem

- A nonlinear problem of the type above was considered by Moles was a biochemical three-step pathway.
- The problem was then used as a benchmark for local and global optimization methods by Moles, Mendes and Banga.
- The problem was then considered by Müller, Lu, Kügler and Engl using methods from control theory.



Benchmark Problem

Examples of the equations are as follows:

$$\frac{dG_2}{dt} = \frac{V_2}{1 + (\frac{P}{K_{i2}})^{n_{i2}} + (\frac{K_{a2}}{M_1})^{n_{a2}}} - k_2 \cdot G_2$$

$$\frac{dE_1}{dt} = \frac{V_4 \cdot G_1}{K_4 + G_1} - k_4 \cdot E_1$$

$$\frac{dM_2}{dt} = \frac{k_{cat2} \cdot E_2 \cdot \frac{1}{.} K_{m3} \cdot (M_1 - M_2)}{1 + \frac{M_1}{K_{m3}} + \frac{M_2}{K_{m4}}} - \frac{k_{cat3} \cdot E_3 \cdot \frac{1}{.} K_{m5} \cdot (M_2 - P)}{1 + \frac{M_2}{K_{m5}} + \frac{P}{K_{m6}}}$$

There are 36 parameters, 2 fixed parameters and 8 ODE variables in this system.

Tikhonov regularization

For the differential equation given previously

$$y'(t) = f(y(t), x, \tilde{x})$$
$$y(0) = y_0.$$

We can define a forward operator

$$F : \ U_x \longrightarrow U_y$$
$$F(x) = y(t)$$

Tikhonov regularization

For the differential equation given previously

$$y'(t) = f(y(t), x, \tilde{x})$$
$$y(0) = y_0.$$

We can define a forward operator

$$F : \ U_x \longrightarrow U_y$$
$$F(x) = y(t)$$

Then we define $x_\alpha^\delta = argmin \, \|F(x) - y^\delta\|^2 + \alpha\|x - x_0\|^2$

This is the x_0-least squares solution to the problem.

Iterated Tikhonov regularization

In the previous slide, x_0 is a guess at the solution you want. Since x_α^δ should be a better solution than x_0 we can look at an iteration:

$$x_0 = 0;$$
$$x_{\alpha,i}^\delta = argmin \, \|F(x) - y^\delta\|^2 + \alpha \|x - x_{\alpha,i-1}^\delta\|^2$$

This process is called Iterated Tikhonov regularization.

Assuming that x_{true} is smooth enough and an appropriate choice of α, we have convergence of iterated Tikhonov of

$$\|x_{true} - x_{\alpha,i}^\delta\| = O(\delta^{\frac{2i}{2i+1}}).$$

Cost Gradient

To solve the Tikhonov problem, we need a minimization algorithm. Many algorithms use the gradient of the cost functional J.

Cost Gradient

To solve the Tikhonov problem, we need a minimization algorithm. Many algorithms use the gradient of the cost functional J.

To estimate the gradient, one way would be to choose $\varepsilon << 1$ and compute

$$(\nabla J(x))_i = \frac{J(x + \varepsilon e_i) - J(x)}{\varepsilon}$$

This requires solving the forward problem 37 times to get 1 estimate of the gradient.

Cost Gradient

Theorem:
Let $y(t) = f(y(t), x, \tilde{x})$, and $F : U_x \rightarrow U_y$ be the forward operator of the ODE. Also let $J(x) = \|F(x) - y^\delta\|^2 + \alpha\|x - x^\delta_{\alpha,i}\|^2$. Then we have

$$\nabla J = \int_0^T (2\alpha(x - x^\delta) + \psi^T f_x)dt,$$

where ψ solves the final value problem

$$\psi'(t) = -f_y^T \psi + 2(y - y^\delta)^T$$
$$\psi(T) = 0$$

Cost Gradient (proof)

To find the gradient of $J(x)$, we start by introducing a Lagrange multiplier ψ, and note that $L(F(x), x) = J(x)$

$$L(y,x) = \int_0^T (y - y^\delta)^2 \, dt + T\alpha(x - x^\delta_{\alpha,i})^2 + \int_0^T \psi^T(y'(t) - f(y,x,\tilde{x})) \, dt$$

$$L(y,x) = \int_0^T (y - y^\delta)^2 \, dt + T\alpha(x - x^\delta_{\alpha,i})^2 + \psi^T y /_0^T - \int_0^T (\psi')^T y \, dt - \int_0^T \psi^T - f(y,x,\tilde{x}) \, dt$$

Cost Gradient (proof)

Taking small variations δx and δy and collecting like terms:

$$\delta L = \int_0^T (2(y - y^\delta) - \psi'(t)^T - \psi^T f_y)\delta y \; dt$$
$$+ \int_0^T (2\alpha(x - x^\delta_{\alpha,i})\psi^T f_x)\delta x \; dt + \psi(T)^T \delta y(T)$$

Therefore if ψ satisfies the final value problem:

$$\psi'(t) = -f_y^T \psi + 2(y - y^\delta)^T$$
$$\psi(T) = 0$$

Then for $y = F(x)$, $\delta L = \nabla J \delta x$, you get

$$\nabla J = \int_0^T (2\alpha(x - x^\delta) + \psi^T f_x) dt.$$

Algorithm

An algorithm for solving this type of problem follows:

1. Choose x_0
2. Apply Tikhonov regularization using the forward problem.
    1. Solve the minimization using a gradient approach
    2. Use the adjoint method to find the gradient
3. Set $x_0 = x_\alpha^\delta$ and iterate.

Conclusion

- We can find the gradient with 2 ODE solves.
- This method gives us an algorithm for finding local solutions.
- Iterated Tikhonov removes the coupling of stability and accuracy.

Conclusion

- We can find the gradient with 2 ODE solves.
- This method gives us an algorithm for finding local solutions.
- Iterated Tikhonov removes the coupling of stability and accuracy.

Future Work:

- Add a global search function and use this to refine to global minimum
- Use other methods and compare accuracy/time results
- Include a parameter selection rule for a larger robustness of code