Inverse Problems, Parameter Estimation, and Optimization

MA 490-01

The Forward Problem

Consider an object of mass m falling under the influence of gravity and air resistance. Take downward as the positive direction, t as time, and $v(t)$ as the downward velocity of the object at time t. The force of gravity on the object is mg with $g > 0$. Let's model the force of air resistance on the object as $-f(v)$, with $f > 0$ for $v > 0$ and $f < 0$ for $v < 0$. From Newton's second law $F = ma$ with $F = -f(v) + mg$ and $a = v'$ we obtain the differential equation

$$mv'(t) = -f(v(t)) + mg. \tag{1}$$

Of course we've made many assumptions (think of some).

If m, g, and the air resistance function f are specified then equation (1), with an initial condition, is a rather standard ODE and is guaranteed to have a unique solution if f is anything reasonable (e.g., if f is C^1). You might be able to solve the ODE in closed form, but more likely you'd have to solve the initial value problem numerically. This situation, in which we know m, g and f, and have to solve equation (1) with some initial condition is called the *forward* or *direct* problem.

The Inverse Problem

A very common situation in the real world is this: We have a mathematical model of some physical situation, e.g., a falling body governed by equation (1), but certain essential parameters in the model are unknown. In the present case that would be m, g, and the function f. Before we can make a quantitative analysis of the model (solve the ODE) we need to either measure the parameters or infer them from experiment or observation.

In the falling body model we could easily measure m and g, but f is more problematic. One way to get a handle on f is to actually drop the object and observe its motion by measuring v at certain times t. From this data we could try to determine the function f. This situation is just the opposite of the forward problem—there we knew f and had to solve for $v(t)$—here we know $v(t)$ (via measurement) and have to determine f. This is called the *inverse* problem. It also goes under the general heading of *parameter estimation*, in which we must estimate the governing physical parameters in some mathematical model.

In general, the forward problem is governed by an ODE or PDE. In the forward problem we have a DE with known coefficients, boundary and initial conditions, and we solve the DE. Conceptually the situation is of the form

$$F := V \to S \tag{2}$$

where V represents the space of allowable coefficient and/or boundary/initial conditions, and S represents the solution space. In the falling body problem V consists of all legitimate choices for f and S consists of possible solution $v(t)$ on some time interval $[0, T]$, or perhaps just finitely many measurements of $v(t)$, at discrete times. Abstractly, the forward problem consists of evaluating F.

The inverse problem involves inverting F, of course. We are told that $F(x) = y$ for some y and we are to recover $x \in V$ which solves this equation.

Depending on the specific forward problem there may be NO choice for x which satisfies $F(x) = y$. In this case we may settle for the x which minimizes $|F(x) - y|$ in some appropriate sense. Or it may be the case that there are infinitely many solutions to $F(x) = y$, in which case we choose an $x \in V$ with some desirable properties (if V consists of functions we might pick the smoothest possible x, or the "smallest" possible x, etc.)

Back to the Falling Body Problem

Physically reasonable choices for f in the model (1) should satisfy $f(0) = 0$, $f(v) < 0$ for $v < 0$, $f(v) > 0$ for $v > 0$, and f should be continuous and strictly increasing. Such function will comprise our space V. With this in mind, suppose the "real" function governing a specific falling body is $f_0(v) = \frac{1}{5}v(e^{v/3} - 1)$. Let's fix $m = 1$ and $g = 9.8$, and take these as known. Solve equation (1) with $f = f_0$ and initial condition $v(0) = 0$. Let the solution be denoted by $v_0(t)$. I did this in Maple, numerically, and computed the solution at times $t = k/10$ for $1 \leq k \leq 10$. The results (rounded to 3 figures) were

$$0.978, 1.94, 2.86, 3.70, 4.45, 5.05, 5.52, 5.85, 6.08, 6.22.$$

These ten data points $v(k/10)$ are what we have to recover f_0.

Thus our space S consists of measurements of the solution to equation (1) at time $k/10$ for $1 \leq k \leq 10$; so S is some subspace of $\mathbb{R}^{10}$. The data above is one particular element of S, and we have to figure out which f gave rise to it.

Least Squares Solution

In our case the y in $F(x) = y$ is an element on $\mathbb{R}^n$. One popular method for solving $F(x) = y$ for x is to minimize the quantity $\|F(x) - y\|^2$, in the sense of Euclidean distance. This doesn't require that we know how to invert F, but only that we know how to evaluate the forward operator F (by solving a DE), and possibly compute its derivatives (how?!).

Before we can proceed, we need to "parameterize" the space V. We will seek an estimate of f_0 which is a polynomial of the form

$$f(v) = v \sum_{k=1}^{n} e^{x_k} v^{k-1}. \tag{3}$$

The reason for using coefficients e^{x_k} instead of just x_k is that we want f to be increasing, we don't want any terms like $-2v^2$; using e^{x_k} prevents this. By the way, I'm not worrying about $v < 0$, since in our problem $v > 0$ at all times. Also, reason for the v out front is that I want automatic enforcement of $f(0) = 0$.

Our goal is then to minimize, as a function of $\mathbf{x} = (x_1, \ldots, x_n)$, the function $\phi(\mathbf{x})$ defined by

$$\phi(\mathbf{x}) = \frac{1}{2}\sum_{k=1}^{10}(v(k/10, \mathbf{x}) - v_0(k/10))^2$$

where $v(t, \mathbf{x})$ denotes the solution to equation (1) with zero initial velocity and $f(v)$ defined by (3). The quantities $v_0(k/10)$ we computed above.

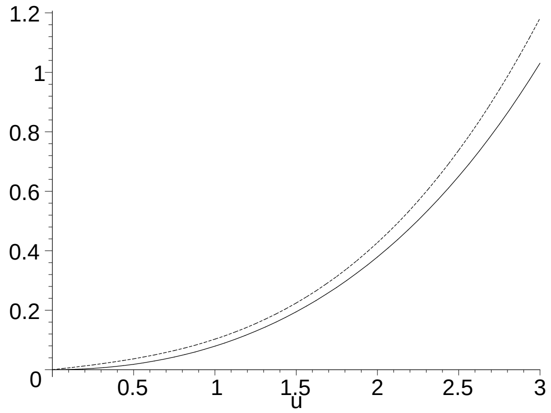
Our hope is that we'll find some magic numbers $\mathbf{x}$ so that $\phi = 0$, i.e., the computed data matches the measured data exactly. Probably we'll have to settle for what we can get.

I used the least-squares notebook `leastsqrs2.mws` with suitable modifications (basically the Levenberg-Marquardt method). One issue is that of computing the gradient; I simply used finite differencing. Whether ϕ is actually differentiable is another issue. Note we don't have a formula for $\phi(\mathbf{x})$, merely a procedure for computing it. However, it is differentiable. Also, if the x_k become too large, either negative or position, then the DE solver has trouble with underflow or overflow. I added penalty terms to the objective function, to penalize large $|x_k| \geq 5$ value for the x_k.

With $n = 3$ the minimizer is $\mathbf{x} = (-2.79, -5.0, -3.36)$, using initial guess $(-1, -1, -1)$, with 199 iterations! The resulting estimate for f is

$$f(v) = v(0.0612 + 0.0067v + 0.0347v^2).$$

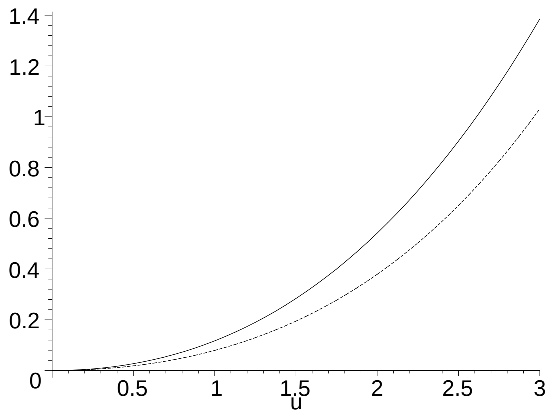
A plot of this function and the original f_0 looks like



The real f_0 is the solid line, the estimate the dashed line.

However, there is some difficulty in the minimization. The whole thing is a bit finicky since large values for $\mathbf{x}$ cause $f(v)$ to grow rapidly and the numeric DE solve to stall. In fact, if $f(v)$ grows too rapidly the DE may not have a solution on the entire interval $0 < t < 1$. I had to play around with the optimization parameters and initial guess to get things to work. Also note that the penalty for $|x_k| \geq 5$ appears to be actively constraining the solution in the x_2 variable.

In reality we don't know the measured data v_0 exactly. To simulate a bit of noise in the data I added to each $v_0(k/10)$ a random error ϵ_k where ϵ_k is normally distributed with mean 0 and standard deviation 0.1. The resulting reconstruction looks like



The function f_0 is the dashed line and the reconstruction is the solid line.

In General

The general setup for using optimization methods to solve inverse or parameter estimation problems is this: Let S be the "data" or "measurement" space. In the example above the space S consists of measuring $v(t)$ for all $t \in (0, 1)$. Of course in reality we can't do that, so we "discretize" S, by replacing S with $\tilde{S}$ where $\tilde{S}$ consists finite dimensional vectors, constructed by sampling elements of S at finitely many points. In the example above $\tilde{S}$ was a subset of $\mathbb{R}^{10}$. We also discretize the space V, replacing it with a finite dimensional approximation $\tilde{V}$, in which general functions are replaced by polynomials or something similar. Thus $\tilde{V}$ is a subset of $\mathbb{R}^m$. In the example above we had $n = 10$ and $m = 3$.

The general approach to solving the inverse problem, given data $y \in \tilde{S}$, is to minimize $\|F(x) - y\|^2$ over $x \in \tilde{V}$. This becomes a finite dimensional least-squares problem, which can be attacked with any standard optimization method.

Pros and Cons

The advantages to this approach are

1. It's conceptually simple and general, requiring only the ability to evaluate the forward operator F and maybe it's derivatives. You don't have to think too hard about the

inverse problem itself.

2. It often works well.

The disadvantages are

1. The optimization formulation of the problem may contain MANY local minima which are not in fact solutions to $F(x) = y$, and so the optimization algorithms may fall into one of these.

2. On large problems it may be too computationally intensive and hence slow.

3. By replacing the "true" spaces S and V with finite dimensional approximations we change the problem, and the solution to the finite dimensional problem may not be an accurate solution to the "real" problem.

4. Inverse problems are often "ill-posed", in the sense that even if there is a unique solution $x = x_0$ to $F(x) = y$ for a given y, other radically different and undesirable choices for x may "nearly" solve $F(x) = y$. In the presence of any noise the optimization algorithm may find one of these undesirable solutions.

5. In many mathematicians' view, the worst objection to this approach is this: The least-squares approach gives no insight into the structure of the inverse problem—questions like "is it uniquely solvable?" "How stable is the solution with respect to noise in the data?" "What is the structure and nature of the inverse operator F^{-1}?" Since mathematicians are into structure and analysis, these are the really interesting questions that the optimization approach circumvents. Also, understanding the inverse problem and F^{-1} can lead to far more efficient algorithms than least-squares.