The adjoint method for IVP-ODE-constrained optimization problems

Andrew M. Bradley November 5, 2008

I describe obtaining the gradient of a function constrained by an initial-value ordinary differential equation by means of the adjoint IVP ODE. As illustration, I apply the technique first to a simple problem whose solution can be obtained in closed form, and then to a nonlinear least squares problem involving an unknown initial condition in addition to other unknown parameters.

1 The adjoint method

Consider the problem

$$\begin{aligned} \min_p \ & \int_0^T f(x,p)\ \mathrm{d}t \equiv F(x,p) \\ \text{s.t. } & h(x,\dot{x},p,t) = 0 \\ & x(0) = x_0(p), \end{aligned}$$

where p is a vector of unknown parameters; x is a (possibly vector-valued) function of time; $h(x,\dot{x},p,t) = 0$ is an ODE in implicit form; and $x(0) = x_0(p)$ is the initial condition, which is a function of some of the unknown parameters. In a data fitting application, the objective might have the form

$$\frac{1}{2}\int_0^T (x(t) - x_{\mathrm{data}}(t))^T (x(t) - x_{\mathrm{data}}(t))\ \mathrm{d}t,$$

where x_{data} is a function obtained by interpolating the data to be fit.

If the problem is formulated such that constraints affect only p, then $h(x,\dot{x},p,t) = 0$ is satisfied at each iteration. A gradient-based optimization algorithm requires that the user calculate the total derivative (gradient)

$$\mathrm{d}_p F(x,p) = \int_0^T \partial_x f \mathrm{d}_p x + \partial_p f\ \mathrm{d}t.$$

(∂_x denotes a partial derivative with respect to x; d_x, a total derivative.) Calculating $\mathrm{d}_p x$ is problematic unless x is available in closed form. If it is not, then two common approaches simply do away with having to calculate it. One approach is to approximate the gradient $\mathrm{d}_p F(x,p)$ by finite differences over p. Generally, the computational effort grows linearly with the number of elements in p, though implementation details, such as data retrieval, may change that estimate. A second method, and the subject of this tutorial, is to develop a second ODE, one in the adjoint vector λ, that is instrumental in calculating the gradient. The primary benefit of the second approach is that the work is approximately equivalent to integrating two ODE.

The first step is to develop the Lagrangian corresponding to the optimization problem:

$$\mathcal{L} \equiv \int_0^T f(x,p) - \lambda^T h(x,\dot{x},p,t)\ \mathrm{d}t - \mu^T(x(0) - x_0(p)).$$

Because the two constraints are always satisfied and so are identically zero, $\mathrm{d}_p\mathcal{L} \equiv \mathrm{d}_p F$. Taking this total derivative,

$$\mathrm{d}_p\mathcal{L} = \int_0^T \partial_x f \mathrm{d}_p x + \partial_p f - \lambda^T(\partial_x h \mathrm{d}_p x + \partial_{\dot{x}} h \mathrm{d}_p \dot{x} + \partial_p h)\ \mathrm{d}t - \mu^T(1-1). \quad (1)$$

First, it is evident that the term containing μ is not used, as its derivative is zero. Second, the integrand contains terms in $\mathrm{d}_p x$ and $\mathrm{d}_p \dot{x}$. The next step is to integrate by parts to get rid of the second one:

$$\int_0^T \lambda^T \partial_{\dot{x}} h \mathrm{d}_p \dot{x}\ \mathrm{d} = \int_0^T \lambda^T \mathrm{d}_t(\partial_{\dot{x}} h \mathrm{d}_p x)\ \mathrm{d}t$$

(because $\partial_{\dot{x}} \mathrm{d}_t h$ is identically zero)

$$= \lambda^T \partial_{\dot{x}} h \mathrm{d}_p x\Big|_0^T - \int_0^T \dot{\lambda}^T \partial_{\dot{x}} h \mathrm{d}_p x\ \mathrm{d}t.$$

Substituting this result into (1) and collecting terms in $\mathrm{d}_p x$,

$$\mathrm{d}_p\mathcal{L} = \int_0^T (\partial_x f - \lambda^T \partial_x h + \dot{\lambda}^T \partial_{\dot{x}} h)\mathrm{d}_p x + f_p - \lambda^T \partial_p h\ \mathrm{d}t - \lambda^T \partial_{\dot{x}} h \mathrm{d}_p x\Big|_0^T.$$

As we have already discussed, $\mathrm{d}_p x(T)$ is difficult to calculate, and we wish to avoid it. Therefore, we set $\lambda(T) = 0$ so that the whole term is zero. In contrast, $\mathrm{d}_p x(0)$ is simple to calculate: it is just $\partial_p x_0(p)$. Similarly, we can avoid computing $\mathrm{d}_p x$ at all other times $t > 0$ by setting $\partial_x f - \lambda^T \partial_x h + \dot{\lambda}^T \partial_{\dot{x}} h = 0$.

The algorithm for computing $\mathrm{d}_p F$ follows:

1. Integrate $h(x,\dot{x},p,t) = 0$ for x from $t = 0$ to T with initial condition $x(0) = x_0(p)$.
2. Integrate $\partial_x f - \lambda^T \partial_x h + \dot{\lambda}^T \partial_{\dot{x}} h = 0$ for λ from $t = T$ to 0 with initial condition $\lambda(T) = 0$.
3. Set
$$\mathrm{d}_p F = \int_0^T f_p - \lambda^T \partial_p h\ \mathrm{d}t + \lambda^T \partial_{\dot{x}} h \mathrm{d}_p x\Big|_0.$$

2 A very simple closed-form example

As an example, let's obtain the gradient of the objective in

$$\begin{aligned} \min_{a,b} \ & \int_0^T x \ \mathrm{d}t \\ \text{s.t. } & \dot{x} = bx \\ & x(0) = a. \end{aligned}$$

The objective is unbounded below and so is senseless; but we are interested only in obtaining a gradient. In this problem, $p = (a \ b)$. We follow each step:

1. Integrating the ODE yields $x(t) = ae^{bt}$.
2. In this problem, $f(x, p) \equiv x$, and so $\partial_x f = 1$. Similarly $h(x, \dot{x}, p, t) \equiv \dot{x} - bx$, and so $\partial_x h = -b$ and $\partial_{\dot{x}} h = 1$. Therefore, we must integrate

   $$\begin{aligned} 1 + b\lambda + \dot{\lambda} &= 0 \\ \lambda(T) &= 0, \end{aligned}$$

   which yields $\lambda(t) = b^{-1}(e^{b(T-t)} - 1)$.
3. In this problem, $\partial_p f = (0 \ 0)$ and $\partial_p h = (0 \ \ -x)$. Therefore, the term

   $$\lambda^T \partial_{\dot{x}} h \mathrm{d}_p x\big|_0 = \lambda(0) \cdot 1 \cdot 1 = b^{-1}(-1 + e^{bT})$$

   gives the first component of the gradient; and the term

   $$\int_0^T \lambda x \ \mathrm{d}t = \int_0^T b^{-1}(e^{b(T-t)} - 1) a e^{bt} \ \mathrm{d}t = \frac{a}{b} T e^{bT} - \frac{a}{b^2}(e^{bT} - 1)$$

   gives the second component.

As a check, let us calculate the total derivative directly. The objective is

$$\int_0^T x \ \mathrm{d}t = \int_0^T a e^{bt} \ \mathrm{d}t = \frac{a}{b}(e^{bT} - 1).$$

Taking the derivative of this expression with respect to a and, separately, b yields the same results we obtained by the adjoint method. (In this problem, of course, the adjoint method is a rather circuitous way to arrive at the answer.)

3 A MATLAB example

Forthcoming.