5 - Adjoint method

Scribed by Deshana Desai, Luca Venturi

1 Adjoint Method

Consider we are dealing with a constrained optimization problem

$$\min_{\mathbf{p}} g(\mathbf{u}(\mathbf{p}), \mathbf{p})$$

where $\mathbf{u}(\mathbf{p}) \in \mathbb{R}^N$ is the solution to some equation

$$\mathbf{f}(\mathbf{u}(\mathbf{p}), \mathbf{p}) = \mathbf{0}$$

If we wish to optimize g using some gradient-type algorithm, we need an efficient way to compute the derivative of g w.r.t. $\mathbf{p} \in \mathbb{R}^M$:

$$\frac{dg}{d\mathbf{p}} = \frac{\partial g}{\partial \mathbf{p}} + \frac{\partial g}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \mathbf{p}} \tag{1}$$

The adjoint method is a general procedure to reduce the computation of $\frac{dg}{d\mathbf{p}}$ to solving one *adjoint equation*.

1.1 A linear example

Consider the case of a linear function g

$$g(\mathbf{u}) = \mathbf{c}^T \mathbf{u}$$

where $\mathbf{u}(\mathbf{p})$ is the solution to a (parametric) linear system

$$\mathbf{A}\mathbf{u}(\mathbf{p}) = \mathbf{b}(\mathbf{p}) \tag{2}$$

where $\mathbf{b} : \mathbb{R}^M \to \mathbb{R}^N$ and $\mathbf{A} \in \mathbb{R}^{N \times N}$ is a fixed matrix. $\mathbf{u}(\mathbf{p})$ is the solution to the system and is thus given by $\mathbf{u}(\mathbf{p}) = \mathbf{A}^{-1}\mathbf{b}(\mathbf{p})$. If we derive $\mathbf{u}$ w.r.t. to $\mathbf{p}$, we get

$$\frac{\partial \mathbf{u}}{\partial \mathbf{p}} = \mathbf{A}^{-1} \frac{\partial \mathbf{b}}{\partial \mathbf{p}} \tag{3}$$

Plugging this into equation (1), we get

$$\frac{dg}{d\mathbf{p}} = \frac{\partial g}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \mathbf{p}} = \mathbf{c}^T \mathbf{A}^{-1} \frac{\partial \mathbf{b}}{\partial \mathbf{p}} \tag{4}$$

Recall that $\mathbf{c}$ is a N dimensional vector, $\mathbf{A}$ is a $N \times N$ matrix, and $\frac{\partial \mathbf{b}}{\partial \mathbf{p}}$ is a $N \times M$ matrix. There are two possible ways to compute the matrix product in the RHS of equation (4):

- $\left(\mathbf{c}^T\mathbf{A}^{-1}\right)\frac{\partial \mathbf{b}}{\partial \mathbf{p}}$. This results in a computational cost of $O(N(N+M))$.
- $\mathbf{c}^T\left(\mathbf{A}^{-1}\frac{\partial \mathbf{b}}{\partial \mathbf{p}}\right)$. This results in a computational cost of $O(NM(1+N))$.

Clearly, the first method is more efficient as $M > 1$. Summarizing, we can re-formulate the above as

$$\frac{\partial g}{\partial \mathbf{p}} = \boldsymbol{\lambda}^T \frac{\partial \mathbf{b}}{\partial \mathbf{p}}$$

where $\boldsymbol{\lambda}$ is the solution to

$$\mathbf{A}^T\boldsymbol{\lambda} = \mathbf{c}$$

This is basically the main idea of the adjoint method. Notice that, in fact, we moved to solving a linear system whose matrix is the adjoint of the original linear system (2). In the following, we see how to apply the same procedure to more involved settings.

1.2 The non-linear case

Consider a generic non-linear function $g : \mathbb{R}^N \times \mathbb{R}^M \to \mathbb{R}$ and assume that $\mathbf{u}(\mathbf{p})$ is taken to be the solution to a generic non-linear system

$$\mathbf{f}(\mathbf{u}(\mathbf{p}), \mathbf{p}) = \mathbf{0}$$

for some $\mathbf{f} : \mathbb{R}^N \times \mathbb{R}^M \to \mathbb{R}^N$. Deriving the above equation w.r.t. $\mathbf{p}$, we get

$$0 = \frac{d}{d\mathbf{p}}\mathbf{f}(\mathbf{u}(\mathbf{p}), \mathbf{p}) = \frac{\partial \mathbf{f}}{\partial \mathbf{u}}\frac{\partial \mathbf{u}}{\partial \mathbf{p}} + \frac{\partial \mathbf{f}}{\partial \mathbf{p}}$$

Solving for $\frac{\partial \mathbf{u}}{\partial \mathbf{p}}$, we get

$$\frac{\partial \mathbf{u}}{\partial \mathbf{p}} = -\left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right)^{-1}\left(\frac{\partial \mathbf{f}}{\partial \mathbf{p}}\right)$$

Plugging this into (1), we get

$$\frac{dg}{d\mathbf{p}} = \frac{\partial g}{\partial \mathbf{p}} + \frac{\partial g}{\partial \mathbf{u}}\frac{\partial \mathbf{u}}{\partial \mathbf{p}} = \frac{\partial g}{\partial \mathbf{p}} - \frac{\partial g}{\partial \mathbf{u}}\left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right)^{-1}\left(\frac{\partial \mathbf{f}}{\partial \mathbf{p}}\right)$$

As before, the efficient way to evaluate the above term is by first computing the product $\frac{\partial g}{\partial \mathbf{u}}\left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right)^{-1}$ and then multiply the resultant matrix with $\frac{\partial \mathbf{f}}{\partial \mathbf{p}}$. Equivalently, we can define the adjoint $\boldsymbol{\lambda}^T$ as the solution to the equation

$$\left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right)^T\boldsymbol{\lambda} = -\left(\frac{\partial g}{\partial \mathbf{u}}\right)^T$$

and take

$$\frac{dg}{d\mathbf{p}} = \frac{\partial g}{\partial \mathbf{p}} - \boldsymbol{\lambda}^T\left(\frac{\partial \mathbf{f}}{\partial \mathbf{p}}\right)$$

1.3 The adjoint method for recurrence relations

Assume we wish to evaluate $\frac{\partial g}{\partial \mathbf{p}}$, where g is given by

$$g = g^n \doteq g(\mathbf{x}^n(\mathbf{p}), \mathbf{p})$$

where $\mathbf{x}^0 = \mathbf{b}(\mathbf{p})$ and $\mathbf{x}^n$ is defined by the iteration

$$\mathbf{x}^k = \mathbf{f}^k \doteq \mathbf{f}(k, \mathbf{x}^{k-1}, \mathbf{p}) \tag{5}$$

The derivative $\frac{\partial g}{\partial \mathbf{p}}$ reads

$$\frac{dg}{d\mathbf{p}} = \frac{\partial g^n}{\partial \mathbf{x}} \frac{\partial \mathbf{x}^n}{\partial \mathbf{p}} + \frac{\partial g^n}{\partial \mathbf{p}}$$

and taking the derivative of (5) w.r.t. $\mathbf{p}$ we get

$$\frac{\partial \mathbf{x}^k}{\partial \mathbf{p}} = \frac{\partial \mathbf{f}^k}{\partial \mathbf{x}} \frac{\partial \mathbf{x}^{k-1}}{\partial \mathbf{p}} + \frac{\partial \mathbf{f}^k}{\partial \mathbf{p}} = \mathbf{f}^k_{\mathbf{x}} \mathbf{x}^{k-1}_{\mathbf{p}} + \mathbf{f}^k_{\mathbf{p}}$$

where we used the notations $\frac{\partial \mathbf{f}^k}{\partial \mathbf{x}} = \mathbf{f}^k_{\mathbf{x}}$ and similar for other variables. Putting the two equations together, we get

$$\begin{aligned}
\frac{dg}{d\mathbf{p}} &= g^n_{\mathbf{x}} \mathbf{x}^n_{\mathbf{p}} + g^n_{\mathbf{p}} \\
&= g^n_{\mathbf{x}} \left(\mathbf{f}^n_{\mathbf{x}} \mathbf{x}^{n-1}_{\mathbf{p}} + \mathbf{f}^n_{\mathbf{p}} \right) + g^n_{\mathbf{p}} \\
&= g^n_{\mathbf{p}} \left(\mathbf{f}^n_{\mathbf{x}} \left(\mathbf{f}^{n-1}_{\mathbf{x}} \mathbf{x}^{n-2}_{\mathbf{p}} + \mathbf{f}^{n-1}_{\mathbf{p}} \right) + \mathbf{f}^n_{\mathbf{p}} \right) + g^n_{\mathbf{p}} \\
&= g^n_{\mathbf{x}} \left(\mathbf{f}^n_{\mathbf{x}} \mathbf{f}^{n-1}_{\mathbf{x}} \mathbf{x}^{n-2}_{\mathbf{p}} + \mathbf{f}^n_{\mathbf{x}} \mathbf{f}^{n-1}_{\mathbf{p}} + \mathbf{f}^n_{\mathbf{p}} \right) + g^n_{\mathbf{p}} \\
&= \cdots \\
&= g^n_{\mathbf{p}} + g^n_{\mathbf{x}} \left(\left(\prod_{i=n}^{1} \mathbf{f}^i_{\mathbf{x}} \right) \mathbf{b}_{\mathbf{p}} + \sum_{i=1}^{n} \left(\prod_{j=n}^{i+1} \mathbf{f}^j_{\mathbf{x}} \right) \mathbf{f}^i_{\mathbf{p}} \right)
\end{aligned} \tag{6}$$

Equivalently, one can define $\boldsymbol{\lambda}^k = g^n_{\mathbf{x}} \left(\prod_{i=n}^{k+1} \mathbf{f}^i_{\mathbf{x}} \right)$. Then $\boldsymbol{\lambda}^k$ verifies the (backward) recurrent relation

$$\begin{aligned}
\boldsymbol{\lambda}^n &= g^n_{\mathbf{x}} \\
\boldsymbol{\lambda}^k &= \boldsymbol{\lambda}^{k+1} \mathbf{f}^{k+1}_x
\end{aligned} \tag{7}$$

and $\frac{dg}{d\mathbf{p}}$ can be evaluated as

$$\frac{dg}{d\mathbf{p}} = g^n_{\mathbf{p}} + \boldsymbol{\lambda}^0 \mathbf{b}_{\mathbf{p}} + \sum_{i=1}^{n} \boldsymbol{\lambda}^i \mathbf{f}^i_{\mathbf{p}}$$

In this case, the adjoint equation (7) is therefore given by a backward recurrent relation.

Question: Compute the computational cost of evaluating $\frac{dg}{d\mathbf{p}}$ using formula (6) (naively) and using the adjoint method. Conclude that the adjoint method is more efficient.

1.4 ODE constraints

We now turn to the real reason why we are looking into the adjoint method: evaluate $\frac{dg}{d\mathbf{p}}$ where g is given by

$$g = g^T \doteq g(\mathbf{x}^\tau(\mathbf{p}), \mathbf{p})$$

and $\mathbf{x}^\tau$ is the solution to the initial value problem

$$\begin{aligned}\dot{\mathbf{x}}^t &= \mathbf{f}(t, \mathbf{x}^t, \mathbf{p}) \\ \mathbf{x}^0 &= \mathbf{b}(\mathbf{p})\end{aligned}$$

at time $t = \tau$. In the following, we use the notation $\mathbf{x}^t_\mathbf{p}$ for $\frac{\partial \mathbf{x}^t}{\partial \mathbf{p}}$ and similar.

The linear case Let's first consider the easier case where $\mathbf{f}$ is a linear function:

$$\begin{aligned}\mathbf{x}(0) &= \mathbf{b}(\mathbf{p}) \\ \dot{\mathbf{x}}^t &= \mathbf{A}\mathbf{x}^t\end{aligned}$$

The solution is given by $\mathbf{x}^t = e^{t\mathbf{A}}\mathbf{b}(\mathbf{p})$. Plugging this into (1), we get

$$\frac{dg}{d\mathbf{p}} = g^\tau_\mathbf{x} e^{\tau\mathbf{A}}\mathbf{b}_\mathbf{p}$$

Similarly to the linear case, we can call $\boldsymbol{\lambda}^T = g^\tau_\mathbf{x} e^{\tau\mathbf{A}}$ so that $\frac{dg}{d\mathbf{p}} = \boldsymbol{\lambda}^T\mathbf{b}_\mathbf{p}$ Notice that now $\boldsymbol{\lambda} = \boldsymbol{\lambda}^0$, where $\boldsymbol{\lambda}^t$ solves the (backward in time) ODE

$$\begin{aligned}\dot{\boldsymbol{\lambda}}^t &= -\mathbf{A}^T\boldsymbol{\lambda}^t \\ \boldsymbol{\lambda}^\tau &= (g^\tau_\mathbf{x})^T\end{aligned}$$

The adjoint equation is therefore given by backward ODE. We can easily see similarity with the case of a recurrence relation.

The non-linear case It was easy to derive the adjoint equation for the case of a linear ODE. What about a more general ODE? For simplicity, consider the case of g given by

$$g = g^\tau = g(\mathbf{x}^\tau)$$

where $\mathbf{x}^t$ solves

$$\begin{aligned}\dot{\mathbf{x}}^t &= \mathbf{f}(\mathbf{x}) \\ \mathbf{x}^0 &= \mathbf{b}(\mathbf{p})\end{aligned}$$

In this case, it's not that trivial anymore to find $\mathbf{x}^\tau_\mathbf{p}$. Motivated by the above linear case, we can think to consider a linearisation of the ODE:

$$\dot{\mathbf{x}}^t_\mathbf{p} = \mathbf{f}_\mathbf{x}\mathbf{x}^t_\mathbf{p}$$

where $\mathbf{f_x} = \mathbf{f}_\mathbf{x}^t$, which we are assuming here to be well approximated by a constant matrix. As we saw above, the adjoint equation is given by

$$\dot{\boldsymbol{\lambda}}^t = -(\mathbf{f_x})^T \boldsymbol{\lambda}^t$$

and $\frac{dg}{d\mathbf{p}} = \left(\boldsymbol{\lambda}^0\right)^T \mathbf{x}_\mathbf{p}^0$. In the general case, simply by analogy, we could try to define the adjoint ODE by

$$\begin{aligned}\dot{\boldsymbol{\lambda}}^t &= -(\mathbf{f}_\mathbf{x}^t)^T \boldsymbol{\lambda}^t \\ \boldsymbol{\lambda}^\tau &= (g_\mathbf{x}^\tau)^T\end{aligned}$$

Turns out that this choice is actually correct, in the following sense. We know, by definition, that

$$\frac{dg}{d\mathbf{p}} = (\boldsymbol{\lambda}^\tau)^T \mathbf{x}_\mathbf{p}^\tau$$

Moreover, the solution $\boldsymbol{\lambda}^t$ to the above ODE satisfies

$$\left((\boldsymbol{\lambda}^t)^T \mathbf{x}_\mathbf{p}^t\right)' = \left(\dot{\boldsymbol{\lambda}}^t\right)^T \mathbf{x}_\mathbf{p}^t + (\boldsymbol{\lambda}^t)^T \dot{\mathbf{x}}_\mathbf{p}^t = -(\boldsymbol{\lambda}^t)^T \mathbf{f}_\mathbf{x}^t \mathbf{x}_\mathbf{p}^t + (\boldsymbol{\lambda}^t)^T \mathbf{f}_\mathbf{x}^t \mathbf{x}_\mathbf{p}^t = \mathbf{0}$$

Therefore $\left(\boldsymbol{\lambda}^t\right)^T \mathbf{x}_\mathbf{p}^t \equiv \left(\boldsymbol{\lambda}^0\right)^T \mathbf{x}_\mathbf{p}^0$ is constant and thus

$$\frac{dg}{d\mathbf{p}} = (\boldsymbol{\lambda}^\tau)^T \mathbf{x}_\mathbf{p}^\tau = \left(\boldsymbol{\lambda}^0\right)^T \mathbf{x}_\mathbf{p}^0 = \left(\boldsymbol{\lambda}^0\right)^T \mathbf{b}_\mathbf{p}$$

which matches the formula for the case of a linear ODE. A computation similar to the previous considered cases proves the greater efficiency of this formula.

1.5 More

The adjoint method can be generalized, in a very similar fashion, to more involved problems. For an introduction, we refer to (Strang, 2007). One could also notice some affinities with backward automatic differentiation, where the use of the chain rule in a backward fashion allows to compute gradients more efficiently.

References

Strang, G.
2007. *Computational science and engineering*, volume 791. Wellesley-Cambridge Press Wellesley.